Universität Zürich[UZH]

Masterarbeit
zur Erlangung des akademischen Grades
Master of Arts
der Philosophischen Fakultät der Universität Zürich

Telling News from Chatter

Automated Multi-Label Classification of 16th Century Documents

Verfasserin/Verfasser: Ismail Prada Ziegler
Matrikel-Nr: 13-737-440

Referentin/Referent: Prof. Dr. Martin Volk

Institut für Computerlinguistik

Abgabedatum: 01.06.2022

Abstract

Automated document classification remains an important field of research to this day. In this work I test different approaches to multi-label classification on the Latin documents in the Bullinger-Digital collection, which contains letters written between 1523 and 1575. I experiment with different methods of preprocessing, feature extraction and classification with a focus on topic models. Best results were achieved using a Correlated Topic Model trained on letters translated into German. While the experiments were only able to produce usable classifications for those labels with a high number of training samples and a clear definition, the insight gained throughout the work provides a foundation for further research into multi-label document classification.

Zusammenfassung

Automatische Klassifikation von Dokumenten ist bis heute ein wichtiges Forschungsfeld. In dieser Arbeit experimentiere ich mit verschiedenen Ansätzen um mittelalterliche Briefe in Latein aus der Bullinger-Digital-Sammlung verschiedenen Themengebieten zuzuordnen, wobei ein Brief auch mehreren Gebieten zugehören kann. Die Briefe stammen aus dem Zeitraum zwischen 1523 und 1575. Ich teste verschiedene Kombinationen von *Preprocessing*-Einstellungen, *Feature Extraction*- und Klassifikations-Algorithmen, wobei der Schwerpunkt auf *Topic Modelling*-Methoden liegt. Die besten Ergebnisse konnten auf Basis eines *Correlated Topic Model* das auf in modernes Deutsch übersetzten Briefen trainiert wurde. Die Experimente erzielten leider nur brauchbare Resultate für diejenigen Themengebiete, die oft in den Trainingsdaten vorhanden waren und klar definiert werden konnten, aber die Erkenntnisse, welche in dieser Arbeit gewonnen werden konnten, bilden eine Grundlage für weitere Forschung im Bereich der *multi-label*-Klassifikation.

Acknowledgement

I want to thank the Bullinger-Digital project for providing me the data this thesis is based on. Special thanks go out to Lukas Fischer, who provided me the translated letters and Raphael Schwitter and Patricia Scheurer for their advice when creating the label set.
Further I want to thank the "Fachstelle Theologenbriefwechsel" from the Heidelberg Academy of Sciences and Humanities and especially Daniel Degen for providing me with their data so I could include them for further research.
Of course I also extend my gratitude to the supervisor of this thesis, Martin Volk, for providing his advice and guidance throughout the project.
Last but not least, I want to thank my friends who proofread the present text.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Heinrich Bullinger was a Swiss reformer who lived from 1504 to 1575. Born in Zurich, he went to study at the university of Cologne and there turned away from the catholic church. He came back to Switzerland in 1523, where he became a follower of the famous reformer Huldrych Zwingli. In 1531, he was named successor to Zwingli as the head of the Zurich church. In this position, he held a lot of influence in the Swiss Reformation movement as well as the city politics of Zurich itself. Not only did Bullinger publish a number of books and essays, but there also remains a big number of letters written by and to Bullinger [Bächtold, 2011].
In these letters, Bullinger keeps contact to various people of his time. Many of them are fellow reformers, mostly from Switzerland and Germany. Some are students of Bullinger, who write him from abroad and others are people that petition him for one thing or another. The information in the letters varies strongly, some are simply there to cultivate relationships, others share news about politics and war, or contain theological discussions.
The letters are mostly kept in Latin, but there is also a share in Early New High German. As Early New High German poses an additional challenge due to the non-standardised orthography and the lack of adapted tools and resources, and only makes up a small part of the total corpus, this work will focus on the Latin texts.
Due to the large number of Bullinger letters, it is tedious for researchers to look through each of them when interested in a particular topic. Keyword searches have their limitations, as often times, relevant people or events are not even named in the letter, but only implied, as Bullinger as well as his conversation partner know who they are referring to.
Through this work I propose automated document classification as one solution to this issue. I put forward that modern machine-learning methods are capable of usefully assigning letters to topics, such that a researcher can simply click on a topic to see which letters it refers to. Of course, enriching the letters with topic information also helps when looking at a letter to immediately get to know what it is about. It

should be noted that each letter may be assigned to any combination of topics.

1.2 Research Questions

In this thesis I will answer the following research questions

1. What automated methods exist to classify the Bullinger texts?
2. Which of these methods work best and how does their use influence the classification quality?
3. What specific problems pose themselves with this document collection and how can they be solved?
4. How can these methods be of use for further research?

1.3 Thesis Structure

In this first chapter, I explain my motivation and the task at hand. In the second chapter, the document collection this work is based on will be introduced. In the third chapter, I will delve into the classification methods and the theory behind them. Then, in the fourth chapter, I will describe the setup for the experiments that have been conducted. The results of these experiments will be examined in detail in the fifth chapter and the usefulness of the classification algorithm will be assessed. Finally, in the sixth chapter, I will present my conclusion and discuss possibilities for further research.

1.4 Previous Research

The topic of document classification on historical documents, especially Latin or Early New High German, is not well researched yet. This is due to the fact that common feature extraction and document classification algorithms as laid out in chapter 3 are not dependent on a specific language.
As explained in Piotrowski [2012], a typical problem for performing Natural Language Processing tasks on historical texts is data sparseness. On the one hand, this is due to spelling variation, on the other, due to the lack of large machine-readable

corpora, especially annotated corpora. To our advantage, Latin is well standardised in the 16th century, so spelling variation is not a problem. There are also some tools and resources available for Latin, such as the CLTK-Toolkit[1] and corpora like the PROIEL treebank, which contains classical texts from roman authors. The corpora for Latin still have to be chosen with caution, as Latin corpora often feature only a small group of authors from a limited timeframe. I will expand on available resources in chapter 2. For the historical German texts, the situation is worse, with no toolkits available and only few annotated resources.
For text classification in general, a lot of research has been conducted. In Kowsari et al. [2019], the authors provide an extensive overview of older and newer text classification methods and summarise the advantages and disadvantages of the different algorithms. I will go into detail about these method in chapter 3.

[1] `http://cltk.org/`

2 The Data

2.1 Documents

2.1.1 Structure of a Letter: Bullinger-Digital

Access to the Bullinger letters was given to me by the Bullinger-Digital project team.[1] The letters of Bullinger-Digital were provided in a custom curated XML-Format. A lot of useful information is provided in addition to the letter texts, such as the date of writing, the authors, the majority language of the text, as well as structural information about the text like paragraph breaks and sentence boundaries. Senders and recipients of the letters were assigned consistent ids over all letters, so matching entities is simple.
The documents stem from two sources. 3'104 of the letters were edited by the institute for reformatory history at the university of Zurich and have been officially published.[2] These letters contain also so called *regests*, which is a short summary of each letters contents in modern German. The other 5'411 letters are also already edited but have not yet received a *regest* and are not officially published. In the latter part of this work I will refer to some of the published letters by their unique ID, which is abbreviated as HBBW (Heinrich Bullinger Briefwechsel).

2.1.1.1 Translated Letters

I received translations for 2070 letters from Latin to modern German. These letters were translated by Lukas Fischer from the Bullinger-Digital project with a translation system specifically created for the Bullinger letters which outperforms Google Translate by more than two BLEU points [Fischer et al., 2022]. Most translation seem of good quality to me, even if some errors remain, the content is usually understandable. Table 1 shows an example of a sentence which has been incorrectly

[1] `https://www.bullinger-digital.ch/about`

[2] Accessible online here: `http://teoirgsed.uzh.ch/`

translated by the Bullinger-Digital system and Google Translate. It is a common error in which all content words in the letter have been correctly translated, but the sentence structure as well as subject and objects have been confused. As will be discussed in chapter 5, due to the way the documents are represented during the classification process, these errors do not seem to decrease the classification quality in a significant way.

Version	Text
Original	Ex Italia missae literae nescio quid de Turcarum imperatore referant.
Bullinger-Digital	Aus Italien wurde ein Brief geschickt, ich weiß nicht, was der Kaiser über die Türken berichtet.
Goole Translate	Briefe aus Italien wissen nicht, was sie als Kaiser der Türken bezeichnen.
My Translation	Ich weiss nicht, was die aus Italien geschickten Briefe dem Kaiser von den Türken berichten.
English	I do not know what the letters sent from Italy report to the Emporer about the Turks.

Table 1: Comparison of the translation quality by an example.

2.1.2 Structure of a Letter: Süddeutscher Theologenbriefwechsel

I was also provided a number of additional documents of similar character to the Bullinger letters by the team of the project "Theologenbriefwechsel im Südwesten des Reichs in der Frühen Neuzeit (1550-1620)"[3], shortened to THBW from here. The THBW serves well to supplement the Bullinger data because the texts are from a similar time, ca. 16th century, from a similar region, Southwest Germany and Switzerland, and from a similar genre, letters between reformers mainly.
The THBW documents were given to me in the TEI-Format[4], an XML-Standard that is well established in the digital humanities and serves to represent physical letters in a digital space while retaining as much information as possible.
While some of the THBW letters also featured some metadata in the XML-files themselves, such as the sender and the recipient and a date, much more complete information is provided by an excel sheet I was given together with letters. In this sheet, the same metadata is provided as is included in the Bullinger letters. The file also contains regests for most of the letters.
As in the Bullinger collection, the text is split into paragraphs, but no sentence boundaries are marked except by punctuation. The THBW metadata also provided their own annotations. I will go into this in detail in section 2.2.2.1.

[3]Documents are accessible online under `https://thbw.hadw-bw.de`.

[4]`https://tei-c.org/`

2.1.3 Standardized Project Format

Because I used files from the different sources described above, I transformed them into a custom format for easier accessability when experimenting on them. I cut away all information not necessary to my project and built the format in a way that made the elements needed for feature extraction easily accessible. An example of the standardised project format can be found in appendix B.
Some elements are omitted if not needed. If no regest was provided for a letter, that element is missing and if the letter isn't translated, no *translated_text*-element is necessary. Paragraph information was also retained wherever possible for future use. The text was at this point not fully preprocessed, but tidied up so it would provide clean text for the preprocessing tools. This included removing XML markup like footnote references, or in case of the TEI files, markup that signified corrected words or expansions of abbreviated words like in this example:
`<expan>An<ex>no</ex></expan> => Anno`
There were also some text elements in the Bullinger letters that needed to be removed which were not marked as XML-Elements. This was the case when for example an editor supplied an information in a regest that was not present in the text (e.g. the text only mentions a lastname, the editor supplies the firstname). Another common case would be bible quotes in the text to which the editors would add a note which psalm it referenced. While both are noted in the same way, put into square brackets, they would need to be handled differently as the first example provided possibly useful information for the task while the second would probably only introduce more noise to the data. I handled this by looking for numbers inside the brackets. If any numbers were inside the brackets, I assumed it to be a reference to either the bible or another letter and deleted it including the contents. Otherwise, I would delete only the brackets and retain the content.

2.1.4 Data Exploration

2.1.4.1 Document Counts

The Bullinger-Digital collection includes a total of 8515 documents. I was provided an automated translation (from Latin to modern German) of 2070 of those documents. The THBW collection provides another 1382 documents. In table 2 the number of documents is given for each language as well as the number of documents that feature a regest, written in modern German. The documents in the "Other/Mixed" category either do not feature any original text, but only a regest, or

they are written in a third language like French or Greek. For the THBW documents, mixed language documents of Latin and German are also included, as the THBW metadata does not provide any information about which language is dominant in the document.
It should be noted that mixing the Early New High German Texts may not work ideally due to the differing editorial standards. For example, in the THBW collection, the word "vnd" and "und" both mean "and", while in the Bullinger collection all such words have been standardised to "und".

Type	Bullinger	THBW
Latin	6613	871
ENHG	1589	463
Other/Mixed	313	48
Total	8515	1382
Regest	3104	1238

Table 2: Number of documents with texts in the respective languages for each document collection. ENHG short for Early New High German.

2.1.4.2 Timeframe

The timeframe of the Bullinger collection spans from 1523, when Bullinger was 19 years old, to 1575, the year of his death. The THBW collection covers more years, from 1517 to 1618. In Figure 1 the distribution of all letters by date is pictured. It is visible that the majority of letters of the THBW collection are from a later year than the Bullinger collection, but there is a substantial overlap. This can also be concluded by comparing the median dates for both collection, which is 1551 for the Bullinger collection and 1561 for the THBW collection. This small difference in dates is probably not significant enough to be a problem when using these data sets together.

2.1.4.3 Authors

Another interesting fact that can prove relevant is the distribution of authors. People have different writing styles, like to use a certain vocabulary and while due to the fact that most authors are scholars of this time they should use similar terminology and styles, our system might detect the differences between authors instead of differences

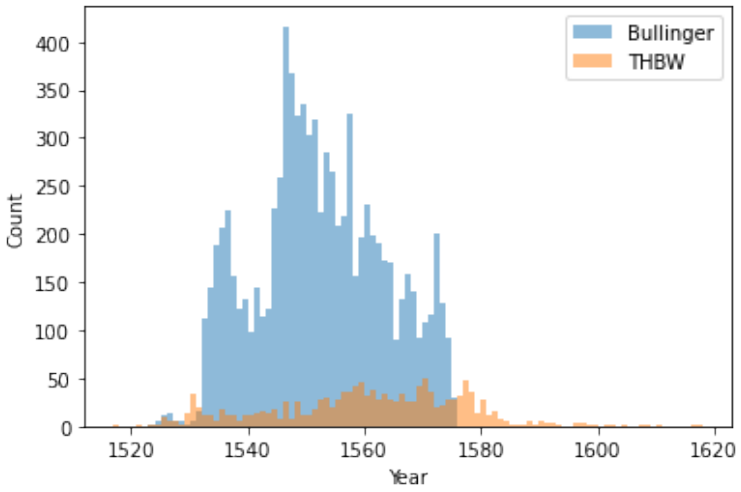

Figure 1: Distribution of letters by year.

between topics in the letters. Interestingly the number of letters by each author follows Zipf's law, with a few people having written most of the letters and roughly half of all senders only having written one letter, in the Bullinger as well as the THBW collection. A total of 840 authors are featured in the Bullinger collection and 213 in the THBW collection. Note the fact that a single letter can have multiple authors.

In table 3 the authors that have written most letters can be seen for each collection. Unsurprisingly, Bullinger wrote most of the letters in the Bullinger collection. The other most common authors there are all reformers that kept a lot of contact to Bullinger. Johannes Haller for example was a reformer from Berne, who had studied in Zurich and was Bullinger's contact in Berne from the late 1540s. Similarly, Blarer was Bullinger's contact in Constance and Myconius in Basel. The authors in the THBW differ significantly. While Bullinger is also present as author of 24 letters, he does not stand out from the others. Instead, Johannes Brenz is the author of almost half of all letters. Brenz was the reformer of the Duchy of Wurttemberg and lived from 1499 to 1570.[5] The other common authors are theologians as well, except the duke of Wurttemberg. Considering Brenz' strong presence as an author in this collection, it is not suprising that his patron would also be featured at least a few times. With Bullinger and Brenz making up a big share of all documents, it is important to check what influence this has on the classification process. The author information can also be used as an additional feature for classification, which will be tested in section 5.8.

[5] `https://en.wikipedia.org/wiki/Johannes_Brenz`

Bullinger	Count	THBW	Count
Heinrich Bullinger	1626	Johannes Brenz	456
Johannes Haller	570	Jakob Andreae	205
Ambrosius Blarer	505	Thomas Erastus	60
Oswald Myconius	346	Duke Christoph of Wurttemberg	48
Johannes Fabricius	277	Petrus Dathenus	43

Table 3: Most common authors.

2.1.4.4 Word Count

Bullinger	Average	Median	SD	Longest	Shortest
Latin	497.4	374	586.6	21560	28
German	474.0	351	444	8429	4
Regest	226.6	134	272.1	4490	5
THBW	Average	Median	SD	Longest	Shortest
Latin	688.4	405	878.1	7933	4
German	906.7	544	1146.9	10288	1
Regest	102.1	81	82.3	719	2

Table 4: Collections compared by word count.

As will be discussed in chapter 3, the word count and the distribution of words can matter significantly in the choice of preprocessing and feature extraction. Table 4 presents information about the word count of the documents. Note that this is counted before any preprocessing and filtering except tokenisation. Figure 2 and Figure 3 visualise the distribution of word count and allows a rough comparison of the Bullinger and THBW collections. A large variance in word count is visible, especially for the Latin letters of the THBW collection. It is also notable that THBW regests are considerably shorter in general than their Bullinger counterparts. Comparing the regests manually, this is very obvious. The THBW regests are written more as a rough summary of the letters content while the Bullinger regests are much closer to the text and are almost summarised sentence by sentence. This is visible as well in the standard deviation values. While the standard deviation is comparatively low for THBW regests, the Bullinger regests also show a high standard deviation, but still less than the original letters.

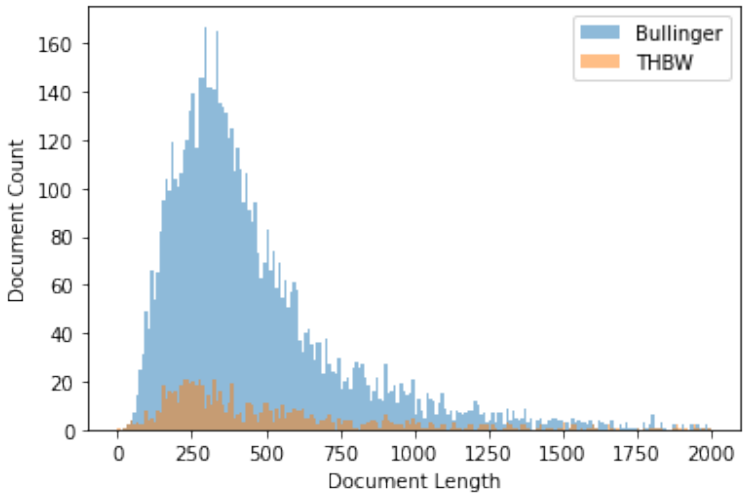

Figure 2: Word count distribution in Latin letters. Documents longer than 2000 words not represented.

2.1.4.5 Word Frequency Distribution

Bullinger	Vocabulary Size	Singletons
Latin	80'380	43'691
ENHG	106'613	70'225
Regest	24'999	11'371
Translation	37'923	19'615
THBW	**Vocabulary Size**	**Singletons**
Latin	33'236	18'070
ENHG	64'500	41'444
Regest	11'285	5'865

Table 5: Vocabulary size and number of singletons per collection.

Table 5 offers an overview of the size of the respective vocabularies (number of word types) as well as the number of singletons (words that only appear once in the collection) in each vocabulary. These counts refer to the words after lemmatisation except for Early New High German, for which no lemmatisation methods exists. As expected, ENHG shows a higher number of words and singletons due to the spelling variation. As will be discussed in chapter 3 the size of the vocabulary and the number of singletons are relevant factors for the quality of the feature extraction process.

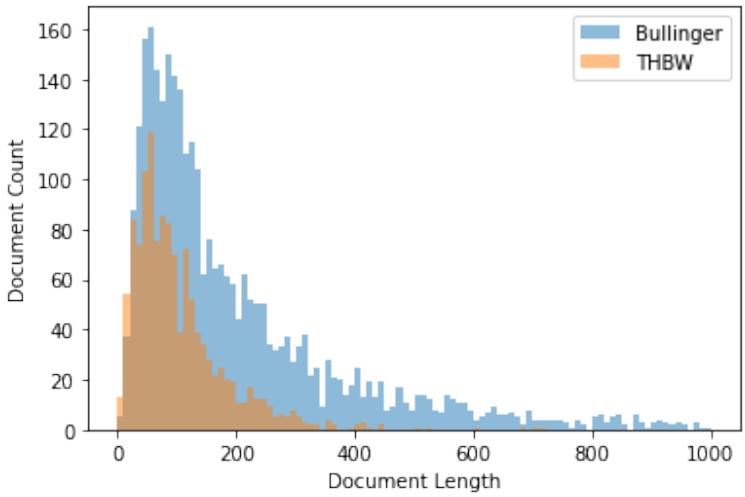

Figure 3: Word count distribution of regests. Regests longer than 1000 words not represented.

The word frequency distribution follows roughly Zipf's law as shown in Figure 4 on the example of the latin Bullinger collection. The other collections feature similar distributions as one would expect.

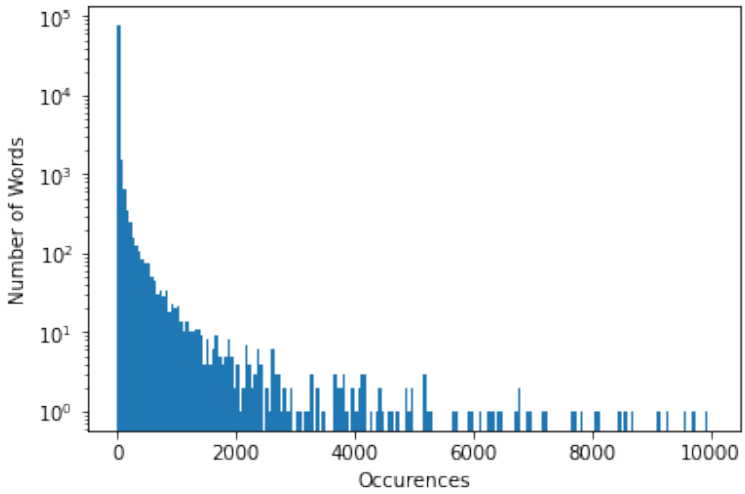

Figure 4: Word frequency distribution on Latin texts. Note that the Y-axis is logarithmic.

2.2 Annotation

2.2.1 Labels

With the stated goal of providing labelled letters for the Bullinger collection, it was essential to assign labels to the letters and paragraphs which would be useful to researchers. I defined a set of useful labels for historians and theologians in collaboration with Raphael Schwitter, a researcher from the department of Greek and Latin philology at the University of Zurich, and Patricia Scheurer of the institute for computational linguistics of the University of Zurich, who are both associated with Bullinger-Digital.
A difficult task was to define labels which would neither be too fine-grained and thus would provide too few examples to be learned effectively, nor be too general, which would make them useless for research purposes. Because the Bullinger-Digital project already has a Named-Entity-Recognition module in work, we tried to avoid designing labels specifically referring to persons or organisations, but obviously some labels correlate with certain persons, for example Luther is often named in texts labelled with *Evangelical-Reformatory Movement*, or places, like a lot of Swiss places will usually be present in letters with the label *Swiss Confederacy*.

2.2.1.1 Label Descriptions

This section will describe the labels we agreed upon and what kind of contents should be labelled with them. The original name of the label given by us in German is given in brackets behind the English translation.

Conflicts (Konflikte): This label is almost always used in combination with others. If used together with *Evangelical-Reformatory Movement* for example, this label refers to theological disputes.

Culture of Correspondence (Briefkultur): This label is assigned to texts which describe how communication was done on a physical level. A typical example would be the request to care for the person who brought the letter, or an excuse that the letter had to be written hastily, because the messenger wouldn't stay much longer. These texts might for example be used in research to find out who usually served as a messenger.

Daily Life (Alltag): All texts which describe events of daily life get this label. An example for this label are reports by students about their accomodation, their travels and their requests for more money or material to continue their studies.

Ecclesiastical Organisation (Kirchliche Organisation): By this label texts are to be found that for example contain information about someone receiving or leaving a clerical office. It can also refer to discussion about how the church should be organised.

Education (Bildung): This label refers to information about student life and organisation of educational institutions, like the search for teachers. It often appears in connection with the Daily Life or Networking label.

Evangelical-Reformatory Movement (Evangelisch-Reformatorische Bewegung): Texts which contain information about affairs inside the evangelical-reformatory movement are assigned this label. A typical example here is letters expressing their opinion about other reformers and their publications as well as information about conflicts between reformers. On a political level, this label might also be given to sections describing public figures embracing the reformation. Abbreviated as *ERM*.

Humanism (Humanismus): In the context of this project, this label refers to texts about the writing, reviewing and publishing of books. Rarely, other scientific discussions are marked with this label, like a letter about an astronomical device.

Islam (Islam): With the Ottoman Empire pushing further into Europe, the Muslim faith and its scripture, the Quran, become of interest to many reformers. Most texts annotated with this label refer to the discussion if the Quran should be translated and published or not.

Military Conflicts (Militärische Konflikte): These labels annotate sections or letters which talk about battles, sieges, but also news about conflicts brewing and troops being summoned by a lord. The same sections are often also labelled with Realm Politics or World Affairs, referring to where the conflict takes place and who is involved.

Networking (Netzwerkpflege): This label marks the text as containing information about the cultivation of the social networks between the writers. It is also assigned when the social network is being used to request something. A typical example would be the sending of a gift mentioned in the letter or the request to recommend a relative to be accepted as a student. Because more or less everything contained in a letter is technically part of cultivating the social network of the author, it is only assigned if the text contains an explicit reference to some act of networking, such as gifting someone something.

Personal Affairs (Persönliches, Familie): Sometimes, letters carry information about family affairs and other very personal information. With this label, they

should be distinguished from other news about daily life or networking.

The Plague (Pest): Numerous texts inform about the plague ravaging a place or someone getting infected with it, these are collected in this label.

Realm Politics (Reichspolitik): Many letters include information about what is happening in the Holy Roman Empire, this label refers to such news. These texts contain information about the struggles between the different political actors of the Empire, be they emperor, nobility, clergy or imperial cities.

Religious Persecution (Glaubensverfolgung): This label refers to information about oppression and persecution of adherents of the reformation.

Roman-Catholic Church (Römisch-Katholische Kirche): This label marks sections of text that contain information about the pope or other news about actors that are part of the Roman-Catholic church. Texts about religious persecutions are not labelled with this label, unless there is more information which refers to the church and its members in other regards.

Swiss Confederacy (Eidgenossenschaft): For information very specifically talking about Swiss affairs, particularly about inter-Swiss conflicts and cooperation, this label is given. Similarly to Realm politics, this label is often used together with other labels to further define the subject the text is about.

World Affairs (Weltgeschehen): This vague label comprises texts which tell news from around the world, particularly outside of the Holy Roman Empire. Examples include the escapades of the English King Henry VIII. and reports about the Ottoman threat to the christian world.

2.2.2 Ground Truth

To assess the success of the automated classification, I annotated by hand a number of letters. I set myself a budget of time and decided to use all letters that I was able to annotate in that time. The letters to annotate were chosen randomly, so as to not introduce any bias in the evaluation process. The disadvantage to this method is that it leaves some categories with far fewer examples than others. I annotated on paragraph-level, meaning the labels on letter level are a union of all paragraph level labels. While I did read the regests to annotate the letters, I also made sure to check the Latin texts in case the regest writers had deemed some information as too unimportant to include in the regest. While this didn't happen often, as the regests are relatively detailed as laid out in section 2.1.4, there were a few times

information was not included in the regest.
With a total of 318 Latin letters, I believe to have attained a usable corpus to evaluate my methods with. A challenge when annotating the letters was the vagueness of the difference of many labels. It was often unclear if a label would fit into *Realm Politics* or the *World Affairs* category, or in both. Another disadvantage was that I annotated part of the data weeks after the first annotations were done, when I considered my ground truth collection to be too small. This could have led to me labelling letters differently than in the first phase of annotation. A big help here would have been a second annotator, who would have annotated the same documents and enabled me to calculate an inter-annotator agreement, which I could have used as an upper ceiling to compare my automated classification results to.

2.2.2.1 Using the THBW Labels

The THBW collection featured its own set of labels. A total of 1240 letters in the THBW collection are annotated with labels, out of which 767 are in Latin and 437 in Early New High German (Others/Mixed is the rest). These labels are far more fine-grained, something we tried to avoid with our custom labels. They function rather as keywords than as categories, with many labels only appearing once in the collection. This also means that individual letters are annotated with many labels, in average 9 labels per letter. The collection features 4'101 unique labels out of which 2'563 labels appear only once in the collection. This numbers only refer to the labels in the files I was given by the THBW team, I could see in the metadata files which featured more letters than those I was given that some of those "singleton" labels were sometimes given to multiple letters. The THBW team also provided me with a text file in which metadata for the labels was defined, including a hierarchy between labels. An example of the hierarchy the label *Plünderung Antwerpens* (Pillage of Antwerp) belongs to:

- Krieg (War)
  - Krieg und Frieden (War and Peace)
    - Achtzigjähriger Krieg (Eighty Years War)
      - Plünderung Antwerpens (Pillage of Antwerp)

There were some problems present with these hierarchies that are probably due to the project still being a work-in-progress. First, some labels just were not included in a hierarchy although they definitely should have been. Here is an example of

some labels with their "parent" labels:

- Türkenabwehr → None
- Türkenangst → Türken
- Türkenhilfe → Türken
- Türkenkriege → Türken
- Türkenpredigt → None
- Türkenüberfall → None
- Türkische Flotte → None

While all these labels are about the "Türken" as the Germans called the Ottomans at the time, only some of them have "Türken" set as their parent label. There is also the issue of a fine-grained label being a mix of two possible parent label. In the above example it was decided to include "Türkenkriege" (Ottoman Wars) in the parent label "Türken" instead of a label related to war like "Krieg". Another problem were decisions that did not make sense to me, such as the inclusion of the "Medizin" (Medicine), which encompasses sub-labels (labels that belong to this parent label) about health and sickness, under the parent label "Universität" (University), a label which also includes sub-labels about specific universities, but unexpectedly no labels about studying.
To make use of these annotations it would not be advantageous to simply use all top-level labels, due to many labels not being included in any hierarchies at all like "Türkenüberfall" in the example above. Filtering out all top-level labels whose sub-labels do not appear more than a defined number of times in the collection would work to get rid of all these "singleton" labels, but would, for example, remove the difference between *University* and *Medicine*. Instead of only using top-level labels, I retained all sub-labels separately that appeared in the collection more often than a set threshold. All top-level labels not reaching that threshold were discarded as well. By using this method with a threshold of 20 minimum occurrences, a total of 152 labels remained. With the most frequent label being "Altes Testament" (Old Testament) with 294 occurrences.
While this work will feature experiments conducted directly on these labels, I also wanted to combine them with our custom label set for the Bullinger collection. I did so by assigning zero, one or multiple labels of our custom label set to each remaining THBW label. Zero labels were assigned when the THBW label would not provide any information about the categories set in the Bullinger label set, such as *Old Testament* or "Gott" (God). An example for a THBW label assigned with mul-

tiple Bullinger labels is "Osiandrischer Streit", a dispute between reformers about a theory formulated by Andreas Osiander. This label was assigned the Bullinger labels *Evangelical-Reformatory Movement* and *Conflict*.

2.2.3 Annotated Data Exploration

	Latin	ENHG	Avg labels
Bullinger	318	3	4.35
THBW	767	437	2.17

Table 6: Count of Annotated Letters And Average labels Per Letter.

Table 6 gives an overview of the number of annotated letters for each collection and language, as well as the average number of labels for each collection. The number of labels per letter in the THBW collection is now much smaller than the average number of labels in the Bullinger collection. It can be assumed that due to the label filtering described in the previous chapter, some annotation information is lost, or that the Bullinger letters feature less monothematic content than the THBW letters, but considering the fact that THBW letters are on average longer than Bullinger letters, this is unlikely.
This relationship can further be explored by investigating the frequency distribution of labels assigned to both collections, visualised in Figure 5. Not fully shown in this graphic, the *Evangelical-Reformatory Movement* label is assigned to 682 letters, more than half of the collection. Taking into account that the THBW collection has almost four times as many annotated letters, this means roughly the same share of letters are annotated as *ERM* as they are in the Bullinger collection. Where the collections really differ is in certain other labels. *Networking* and *Culture of Correspondence* are highly underrepresented in THBW (or overrepresented in Bullinger). This is not unexpected, as these labels represent typical parts of many letters, which were probably often not annotated in the THBW collection due to the fact that they are so common. On the other hand, the labels *Islam* and *Ecclesiastical Organisation* are much more common in the THBW label set.
Observing only the label distribution of the Bullinger letters, a very strong imbalance between label frequencies is visible. While *ERM* and *Networking* are assigned to more than half of all letters, *Islam* is only assigned to three letters. This imbalance can have adverse effects on the classification algorithms, as will be described in chapter 3. The THBW collection while itself also being similarly imbalanced, with the least frequent label *Islam* only occurring 20 times, may help with this by

offering more samples for low frequency labels in the Bullinger collection, such as *Ecclesiastical Organisation*.

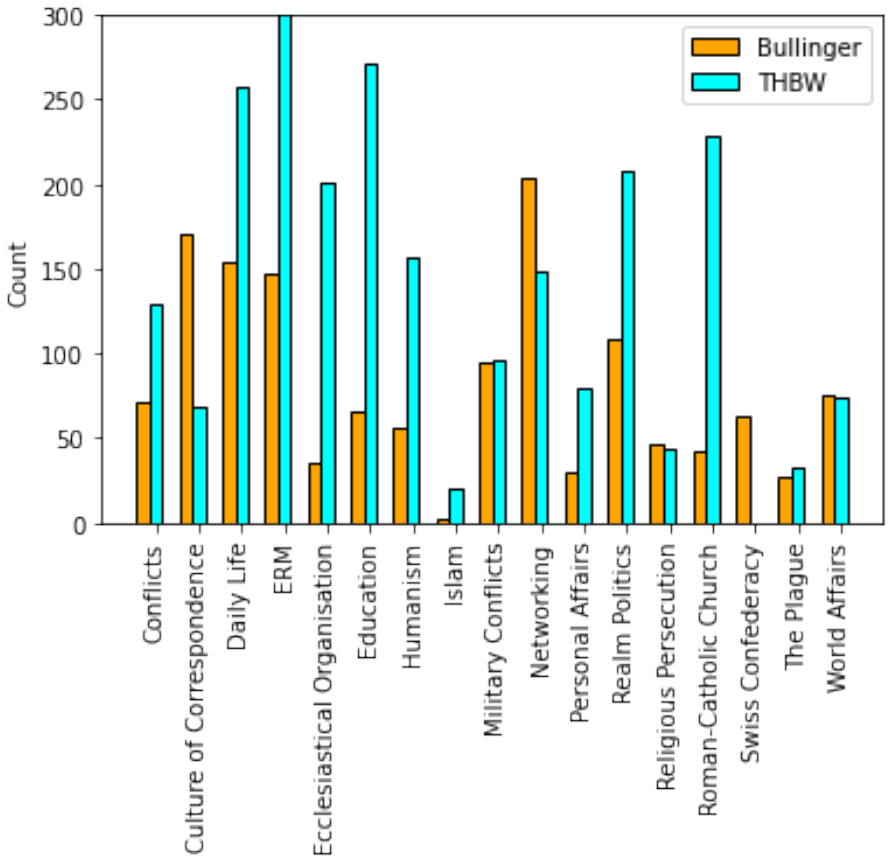

Figure 5: Frequency of labels assigned. ERM not fully represented for THBW as it has a count of 682.

2.2.3.1 Representativeness

Is this small sample of the Bullinger collection, that will be used as test data, representative for the corpus as a whole? The time frame of the annotated Bullinger letters spans from 1534 to 1547, which only covers a small part of the time frame of the whole corpus, which features most letters around 1550. A difference in authorship is also visible with 59 letters written by Oswald Myconius, compared to Bullinger who only wrote 52 of the 321 annotated letters. Comparing these numbers to those observed in section 2.1.4.3 (Bullinger 1626, Myconius 346) leads to the conclusion that Myconius is overrepresented roughly by a factor of 5 in the annotated data set. There are also only 68 authors present out of the 840 authors in the complete collection. The median document length is at 411 words, which is considerably higher than the 374 words median length of the complete Latin Bullinger collection.

These differences are probably caused by the fact that when the letters were annotated, I was working with a smaller collection of documents, which only featured letters from that time frame, in which Myconius is overrepresented as an author. In conclusion, due to the differences between annotated data and the complete collection, the algorithms in this work should be expected to work worse on the complete collection if only trained on the annotated data.

2.2.3.2 Author-Label Correlations

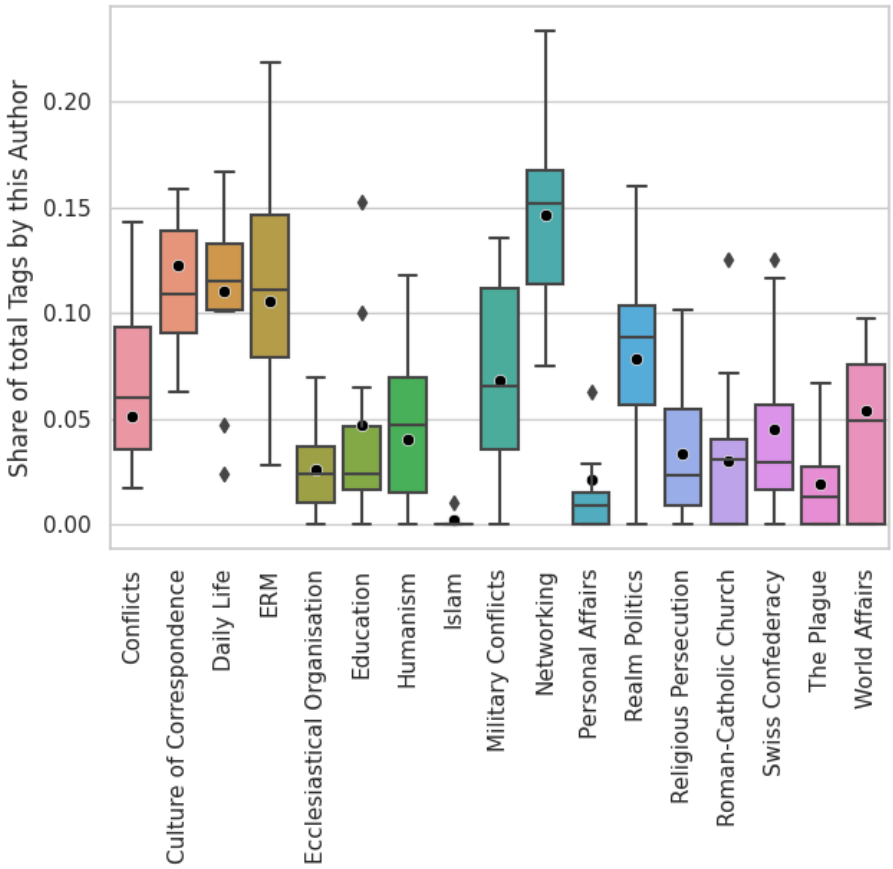

Figure 6: Each data point represents the percentage of the label specified on the X-axis compared to all the labels in the letters by a specific author. Authors with less than 5 letters excluded. Black points indicate the share of that label compared to the all labels.

In this section, we will explore the relationship between authors and labels. Figure 6 visualises how far authors stray from the usual share of a certain label in the corpus. The black point represents the share of a label compared to all labels. In case of all labels, the majority of authors is concentrated around the usual share, but some labels feature a broader distribution, like *ERM*, or a dense distribution with some outliers like *Education*. Especially these outliers, authors write a lot about a

certain topic, like Rudolf Gwalther, who is the outlier at the top in the *Education* label, could be useful features for the classification process. This is understandable considering Gwalther was Bullingers foster son and wrote to him regularly about his studies in the annotated letters. Of course this could also lead to overfitting, meaning the classification would overvalue Gwalthers authorship as feature and only label letters by Gwalther as *Education* or all of Gwalthers letters as *Education*. It should of course be noted that only a small share of all authors are present in the annotated data, so using this information for classification would be difficult to extend to the complete data set unless training data is expanded to include significantly more authors. The usefulness of adding author information as features will be explored in section 5.8.
Doing a similar visualisation with dates instead of authors did not yield any satisfying conclusions due to the small size of the annotated data set.

2.2.3.3 Label Co-occurrences

Knowing which labels often appear in the same letters can be valuable information when later investigating wrong classifications done by the system. In Figure 7 the Pointwise Mutual Information (PMI) values between labels are shown for the annotated Bullinger letters. Pointwise Mutual Information is higher if labels occur often together and do not appear often in general. The label *Islam* skews these results because it appears only 3 times in the collection, so Humanism, itself a rare label, appearing only once or twice with it will already result in a very high PMI. Disregarding the pairs that feature *Islam*, the highest PMI is visible between *Religious Persecution* and *Roman-Catholic Church*, *Military Conflicts* and *World Affairs* and *Ecclesiastical Organisation* and *Roman-Catholic Church*, all combinations that one would expect to co-occur.

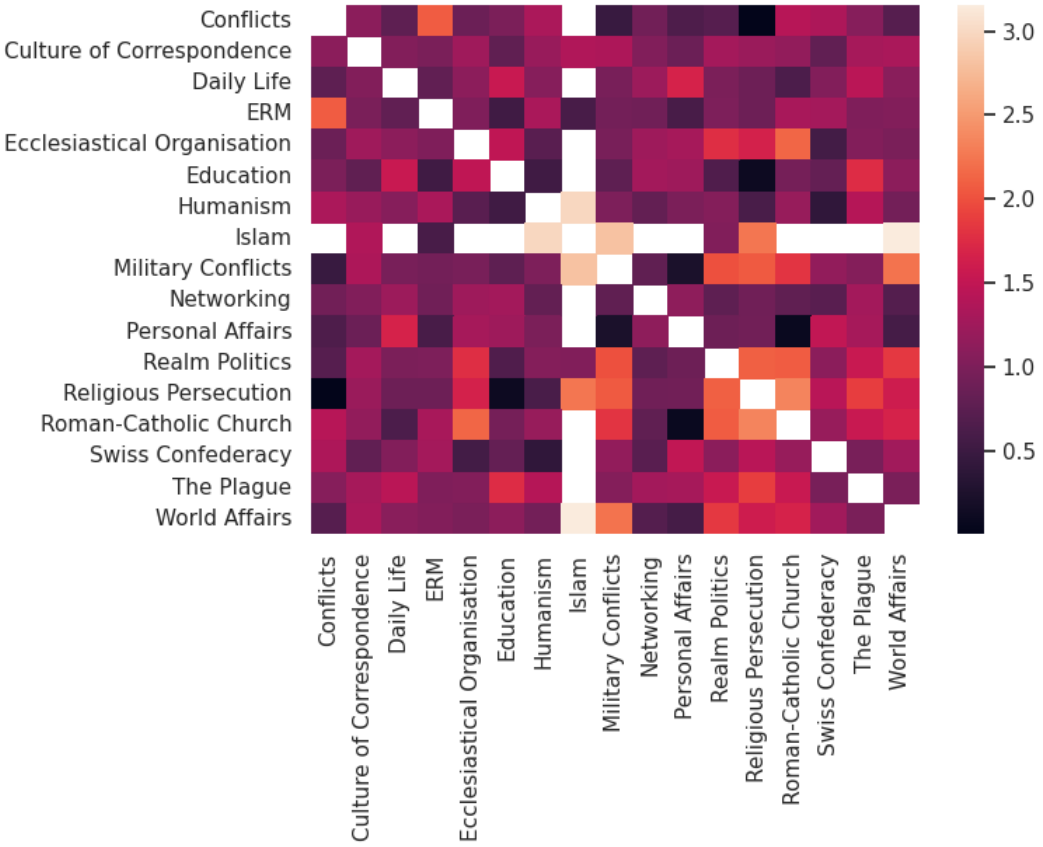

Figure 7: Heatmap based on PMI values between labels (Bullinger collection). A lighter color represents a high PMI.

3 Methods

In this chapter, the different steps and methods employed in document classifications are presented as well as the theory behind them.

3.1 Steps of Document Classification

The process of document classification can be split into three parts:

1. How to preprocess the documents?
2. Which algorithm to use to extract features?
3. Which algorithm to use as the classifier?

3.2 Preprocessing

The step of preprocessing serves to prepare the documents for further processing. The steps described in the following sections have been proven to often improve system performance [Kowsari et al., 2019, 4].

3.2.1 Tokenization

By tokenizing the texts, words are split from each other. While this process is simple, when it comes to white spaces, it is important to perform it especially to separate punctuation from words. This is a challenge because we need to distinguish punctuation that marks sentence boundaries from punctuation that is part of abbrevations. While this is a much rarer case in the Latin texts this project works with, it does still happen. Two extremely common abbreviations in the Bullinger collection are ”s.” for ”salus” (”greetings”) and ”d.” for ”dominus” (”master”, as an address of respect used like ”sir”).

3.2.2 Capitalization

Latin poses similar problems to English when it comes to capitalisation. Only proper names should be in upper case, but sentence starts pose a challenge here. Identifying period signs can help, but sometimes a period might mark an abbreviation. The common way to handle this is to lowercase the whole text [Kowsari et al., 2019, 4]. This might lead to confusion between two different words that can only be distinguished by capitalisation. The most likely case would be with persons last names which could also mean a job. The most likely person to be confused in this way in the Bullinger collection is the theologian Johannes Gast, as the last name could also mean "visitor". Part-Of-Speech tagging can help to mark this distinction in the text, for example to label all named entities with an appendage of "_NE", but because these problems are so rare, no action is taken in this project to make this distinction.

3.2.3 Lemmatisation

Lemmatisation describes the process of transforming a word into its basic form. In case of nouns, this is usually the nominative singular, while for verbs it is the infinitive form. This serves to reduce the number of word types, which supports the feature extraction algorithms.

3.2.4 Stop Words

Like lemmatisation, the removal of stop words, that is, words that do not carry information about the content of the document, reduces the number of words and word types, speeding up the process and improving the performance. But one can do more damage than good when removing stop words if it is not carefully evaluated what method to use to build the list of stop words. In Saif et al. [2014] the authors observe that the classical method of using precompiled stopword lists actually lowers performance in their experiment. This highlights the relevance that even non-important seeming words can have when it comes to certain tasks. When it comes to classification tasks like in this work, there might be the case of a higher number of certain pronouns appearing for certain categories, for example.
Furthermore the removal of singleton words or low frequency words can also be counted among the practice of removal of stop words. As the methods to determine what is a stop word differ for high frequency and low frequency words, I will distinguish between the two in the following chapters. The removal of low frequency

words can be seen in [Saif et al., 2014] as well as [Tang et al., 2014] where the practice of removing words that only appear in few documents is described as common practice before topic modelling is employed.

3.2.5 Noise Removal

Similarly to stop word removal, noise removal reduces the number of word types and words in general by removing punctuation and special characters and thus speeds up and improves performance.

3.3 Feature Extraction

As classification algorithms cannot be trained on textual data itself, it is necessary to convert the documents into a numerical representation. The methods which are tested in this thesis are introduced in this section. While not all of these methods could be included in the experiments, due to missing implementations or time constraints, I include them in this chapter to provide a more complete overview of methods.

3.3.1 Term Frequency-Inverse Document Frequency (TF-IDF)

This method of representing documents is based on the assumption that a term is more meaningful to a document if it (1) appears often in that document and (2) does not appear often in other documents. The current implementations are still based on the paper by Spärck Jones [1972], which was originally published in 1972. The *term frequency* is the number of times a word occurs in a document. The *inverse document frequency* is the number of times a word occurs in the document collection divided by the number of documents the word appears in. The latter number is usually increased by one to prevent zero-division errors if a word does not appear at all in the document collection. The *TF-IDF* for word in a document is then calculated as the product of the term frequency and the inverse document frequency. If both values are high, so is the product. Thus, a more meaningful term gets a higher TF-IDF value assigned to the inspected document.
The algorithm has since been expanded upon and a number of augmentations have been proposed. One such augmentation is sublinear term frequency scaling. The motivation behind this augmentation is that while a term that appears twenty times

more often in a document than another term surely has a higher importance, it is unlikely that that difference is actually that high. By normalising the raw count by using the logarithmic value and adding 1 to it, a term that appears for example 20 times more often than another only gets a 2.3 times higher weighted term frequency than a term that only appears once [Manning et al., 2008].
Another problem that is also relevant for the Bullinger letters in this project, is that the term frequency in its original form does not take into account the length of the document. In collections with a strong variance in document lengths, this is likely to decrease the quality of the TF-IDF results as longer documents automatically get higher TF-IDF scores for most words. Different methods have been proposed to handle this challenge, one of them is applying the euclidean norm on the resulting feature vector.[1]
When using TF-IDF, the feature vector for each document consists of the TF-IDF scores for all words in the vocabulary. Each document of the collection uses the same vocabulary in the same order, so the vectors of two documents where the same words are important will be similar to each other.

3.3.1.1 Dimensionality Reduction

Due to the large vocabularies, usually in the tens of thousands, it is recommended to perform a method of dimensionality reduction on vectors created by TF-IDF. Large vectors take a longer time to train and contain a lot of unnecessary information. Imagine a vocabulary, in which singletons (words that only appear once in the corpus) are included, according to Zipf's law, many of the words in the vocabulary will be singletons. This means for the feature vectors that are calculated, that many features in the document vectors will be 0, except for the one document that singleton appears in. All these features are useless for the next step in the pipeline, the classification, as they will not contribute to find similarities between documents. Even if singletons are excluded, most words will still have score near zero for most documents.
A common way to get rid of these unimportant features is Principal Component Analysis, a method that aims to reduce the vector dimensionality by using Singular Value Decomposition [Abdi and Williams, 2010]. Neural networks have been applied for dimensionality reduction with success as well in the form of autoencoders [Wang et al., 2014].

[1] For example done in the TF-IDF implementation of sklearn: `https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction`

3.3.2 Word Embeddings

One disadvantage of TF-IDF is that it cannot make use of relationships between words. If we encounter the words "army" and "troop" in a vocabulary, they carry a similar meaning to us. But the term frequency will be counted separately for each of them.
Word embeddings are able to represent these relationships. When representing words by their word embeddings, each word has a position in an x-dimensional space. Closer words are related to each other, the closest words usually are synonyms of each other or deviate only by its gendered form.
There are multiple algorithms to train word embeddings, the two most popular being word2vec [Mikolov et al., 2013] and Fasttext [Bojanowski et al., 2017]. Both rely on a shallow neural network that is trained to predict a word based on its context. Fasttext expands on word2vec by integrating information about the word itself, the character n-grams that it contains, into the process. This enables Fasttext to handle out-of-vocabulary words when confronted with new data. Additionally words with low frequency can better be embedded as long as they have similar character n-grams as other words. While out-of-vocabulary words are no issue to this project, because the model can be trained on all available training data, using FastText can still be advantageous to handle low frequency words, especially singletons.
Both algorithms offer the Continuous Bag-Of-Words (CBOW) and the Continuous Skip-Gram (SG) method to train with. When using CBOW, all words in a set-width context of the target word will be considered when predicting the word. Skip-Gram turns this process around and tries to maximize prediction quality for surrounding words based on a given word in a context.

3.3.3 From Word Vectors to Document Vectors

The question remains how to get from word vectors to vectors that represent whole documents. An intuitive way would be to simply average the word vectors of the document. But as Le and Mikolov [2014] point out, this leads to a loss in information. An averaging of words would lose the information about the word order in the document. Another approach, concatenating the word vectors by an order given by a parse tree, only works on sentence level.
In their paper, Le and Mikolov propose *Paragraph Vector*, now better known as Doc2Vec, which expands on the existing word2vec architecture. In their approach, they jointly train word and paragraph vectors. Only at test time, they freeze the word vectors and train the paragraph vectors by gradient descent. This method

of document representation was then evaluated on a sentiment analysis task. The paragraph vectors were used to train a classifier, in their experiments this was done through a shallow neural network for one task and with logistic regression for another task, which was in both cases able to significantly outperform all previous approaches like averaging word vectors or using parse trees.
In [Joulin et al., 2017], Fasttext is tested for use in text classification. They choose the more common method to calculate document vectors, taking the average of the word vectors. The document vectors are then fed to a linear classifier, or in case of many classes, a hierarchical classifier. This simple architecture already yielded results that were on par with more time-intensive deep-learning classifiers. An additional boost in quality could be gained by adding character bigram information to the mix. The authors propose that adding character n-gram vectors to the word vectors when calculating the document vector preserves some information about word order and thus improves the document representation. The authors could observe an increase of 1-4% in performance when adding character bigram information.

3.3.4 Topic Modelling

Topic modelling is a popular method to create document representations without requiring labelled data. The assumption of topic models is the following: Each document consists of a mixture of topics. Each word in a document has a stronger or weaker relation to a certain topic. Topic modelling is, like TF-IDF and word embeddings, an unsupervised process. While the number of topics can be set, the topics itself are not labelled, and the best way for a human to observe what a topic is about, is to look at the top words for each topic or at the documents with the strongest relation to that topic. But the topic probabilities for each document can be used as feature vector to train a classifier, making the topic modelling a step in a supervised task. The idea here is that each one of the labels defined in this project can be represented as a mixture of topics identified by the modelling process. For example the label *Realm Politics* could be a mixture of a topic which contains political affairs and a topic which contains geographical information about the Holy Roman Empire. Of course the ideal case would be a topic exactly matching the label.

3.3.4.1 Probabilistic Latent Semantic Analysis (pLSA)

While pLSA is not tested in this work, I include a section about it here to explain the motivations and ideas behind the following topic modelling algorithms.
Probabilistic Latent Semantic Analysis was the connecting step from TF-IDF to topic modelling, as Latent Semantic Analysis was already used for dimensionality reduction. Hofmann proposed his concept of Probabilistic Latent Semantic Analysis in 1999 [Hofmann, 1999]. His model is the first to represent documents as a mixture of topics, even though the original paper does not use that term but instead "factors". Hofmann's model assumes a data set generated like this:

1. For every topic z sample a word distribution β_z.
2. For every document d sample a topic distribution θ_d.
3. For every document d generate words W:
    a) Sample a topic $z_{d,i}$ from the topic distribution θ_d.
    b) Sample a word $w_{d,i}$ from the word distribution $\beta_{z_{d,i}}$

This relationship between words, topics and documents, Hofmann formulates as:

$$P(d, w) = \sum_{z \in Z} P(z)P(w|z)P(d|z) \tag{3.1}$$

Where Z is a collection of all topics. This model assumes a conditional independence between the document d and the word w, both being connected by the topics Z. The goal of the training process of the pLSA model is thus to optimize the topic distributions for each document and the word distributions for each topic, so the generated data by the model is as similar as possible to the original data. This is done with Maximum Likelihood Estimation using Tempered Expectation Maximization. It should be noted that this process does not take word order into account. After training is finished, the resulting topic distributions for each model can be used as feature vectors.

3.3.4.2 Latent Dirichlet Allocation (LDA)

Hofmann's implementation carries with it some limitations. One of them being that due to topic distributions being handled as parameters for each document, it is not possible to calculate topic distributions for documents that were not in the training corpus. This behaviour also causes the number of parameters to grow linearly with

the number of documents, which causes slow training and makes the model prone to overfitting. The Latent Dirichlet Allocation model proposed by Blei et al. [2003] aims to solve that issue and further improve the topic modelling process by handling the topic-document relationships not as a set of parameters but instead as a single hidden random variable.
The assumption concerning the creation of a corpus is very similar and goes as follows:

1. For every topic z sample a word distribution β_z.
2. For every document d sample a topic distribution θ_d from a Dirichlet distribution with a constant parameter α.
3. For every document d generate words W:
    a) Sample a topic z_i from the topic distribution θ_d.
    b) Sample a word w_i from the word distribution β_{z_i}

It is only a small difference, in this list, but a big one concerning the performance of the model: The topic is now independent from the document and only consists of its word distribution. This enables LDA to also calculate topic distributions for documents not initially part of the training corpus and it means that the model training process no longer scales linearly with the number of documents. The other change, from which of LDA takes its name, is that the topic distribution is now sampled from a dirichlet distribution. Like pLSA, an Expectation Maximization algorithm is used to tune the parameters, in this case the dirichlet parameter α and the word distributions β. Like pLSA, the topic distributions may then be used as feature vectors for classification, which can be calculated for unseen documents as well. The latent dirichlet allocation outperformed pLSA in the task to predict held-out documents (using a method to make pLSA able to predict unseen documents) and demonstrated that it is not prone to overfitting in the same way. Blei et al. [2003] also demonstrated how their model can be used for classification tasks to extract feature vectors and outperform simple word features.

3.3.4.3 Limitations of Topic Modeling

In their 2014 paper, Tang et. al. perform an in-depth analysis on how document count and document length influence the quality of topic models created by LDA [Tang et al., 2014]. They come to five important conclusions:

1. The number of documents is critical to the quality of the topic model. But

when a sufficient number of documents has been reached, increasing the number further will not increase the quality further, unless document length increases as well.

2. The length of the documents is also very important. Documents that are too short hurt the quality of the model the most, but depending on the genre, so can documents that are too long.

3. Choosing too many topics for the model may not only slow down training, but also hurt the model quality.

4. Best quality can be achieved if topics are well separated and concentrated on few words.

5. If each document only contains a few topics α should be set to a small value.

To put forward some numbers, model quality usually stops increasing after a document length between 100 and 200 words has been reached. After this length, the quality even starts to drop for the Twitter documents. The peak for document count varies strongly with the genre. In Tang et al. [2014], many more Twitter documents[2], almost 4000, are needed to reach best quality while Wikipedia articles only improves little after a few hundred articles. This might be attributed to the stronger topic mixture in the Twitter documents compared to Wikipedia articles. Considering that more documents than necessary do not show averse effects in any of the tested document collections, for this work, it can be considered helpful to use as much data as is available.
The average document length in the Bullinger collection is far above what Tang et al. have found to be the minimum recommended document length with a median length of 374 words per document (for Latin texts). After preprocessing by removing stop words by the methods described in chapter 4 the median is lowered to 71 words per document. This is below the mimimum recommended by Tang et al., which is why the impact of filtering out words and thus lowering the document length will be observed in chapter 5.
The Bullinger letters also contain a strong mix of topics in each document, which is why I will also experiment with higher values of α in chapter 5.

[2]Tang et. al. use simple pseudo documents comprised of all tweets by one user to compensate for the shortness of tweets.

3.3.4.4 Variants and Improvements of Latent Dirichlet Allocation

Plenty of research has been done to build upon the architecture that Blei et. al. proposed in 2003, some of it to alleviate the limitations described in the previous section. Newman et al. [2009] introduce distributed algorithms to make use of multi-processor machines and manage to achieve a significant time reduction to train topic models while the models still perform similarly to single-processor architectures.
The problem of short texts has been taken on as well, for example in [Zuo et al., 2016]. In their generative model the topic distributions are sampled for each pseudo document instead of original documents, and each short text is assumed to be part of one pseudo document, from which it gets the topic distribution. The pseudo document for each original document is chosen from a multinomial distribution which is in turn sampled from a dirichlet distribution. Zuo et. al. also describe their topic models as combined topics which can generate the specific topics for the short texts. Thus the pseudo documents can be viewed as an additional layer that separates topics from the original documents. It's assumed this helps because when short texts are used for topic modelling, word co-occurrences are too few to calculate usable topics. By having pseudo documents that are influenced by their short texts, they can provide a higher word count and with it more word co-occurrences, which in turn improves the model quality. This architecture manages to outperform LDA on all and comparable methods to create pseudo documents for topic modelling on most tasks. It should be mentioned that the experiments in this paper were performed on far shorter documents than this project deals with, ranging from an average document length of 4.6 to 12.4.
Blei himself proposed an improvement to his LDA architecture in [Blei and Lafferty, 2005]. LDA does not take into account that some topics in the model might be closely related and that relationship could be used to improve the model quality. In our data, for example, a letter about warfare is much more likely to also be about politics than to be about theological debates (See section **??**). Blei and Lafferty solve this by drawing the topic distributions from a matrix that represents correlation between topic models instead of the dirichlet distribution. This method worked well, performing better at predicting held-out documents, especially if the held-out documents contained previously unseen words, as these could be better inferred through the correlation of topics. This architecture introduces a new problem: The way the topic correlation matrix works leads to a cubic time complexity growing with the number of topics. He et al. [2017] solve this problem by representing the topics in a x-dimensional vector space where closer topics are stronger correlated. This not only reduces the time complexity to a quadratic growth with the number of topics, but also improves quality of the topic model.

Some experiments make use of labelled data in topic modelling. Ramage et al. [2009] proposed an architecture called Labelled LDA (L-LDA) in which each topic corresponds to one of the labels in the dataset. Because each topic would correspond to one label, this model is able to be used for classification directly, not only to provide the features. The main difference in the generative process is in this case that the α parameter used to sample the topic distributions is specific for each document and dependent on the document labels. The tested models performed significantly better at multi-label classification compared to Support Vector Machine classifiers trained on TF-IDF features. Ramage et al. [2011] identify multiple problems with the strict attachment between topics and labels in L-LDA. For example, by using only one topic for each label, it is impossible to identify sub-topics. In their architecture called Partially Labeled Dirichlet Allocation (PLDA), they attempt to combine the best of LDA and L-LDA. In the generative model, this means that now it is assumed that each label is assigned a number of topics (a parameter that can be set by the user) and that each word in a document is generated by first choosing a label, then a topic based on that label and then a word based on that topic. PLDA also introduces an optional latent topic class, which can be seen like a label, which all documents in the data share. This architecture also enables the training with documents without any labels at all. If no labels are provided at all, PLDA works exactly as LDA would. If the number of topics for each label is set to 1 and no latent topic class is used, PLDA works exactly as L-LDA would. PLDA does not only prove to be considerably better performing at predicting labels than pure LDA or L-LDA, but also significantly improves training time.

A lot of new research in topic modelling attempts to make use of word embeddings in topic models. In all previously described models, we still use singular word types and do not consider relationships and similarities between words. It should be mentioned that topic modelling does somewhat take this into account, as words that are similar will usually end up in similar topics, but this does not help when working for example with words that only appear few times in the document collection. In [Dieng et al., 2020], to which also the original LDA creator David Blei contributed, word embeddings were successfully incorporated into LDA. When creating word embeddings, CBOW specifically, the likelihood that a specific word is generated is dependent on the embedding matrix which contains the embedding representations of all words in the vocabulary and the context of the word. When generating words for the documents in this topic model, instead of generating the word based on its context, it is generated based on the sampled topic. The topics themselves are also represented as embeddings existing in the same semantic space as the word embeddings. The word embeddings can either be learned during topic modelling, or can be provided pre-trained word embeddings, in which case only the topic embeddings are

learned. This model is called Embedded Topic Model (ETM). ETM creates topics of a better quality than LDA and previous approaches to integrate word embeddings into topic modelling.
Another approach to integrating word embeddings was taken in [Das et al., 2015] and [Xun et al., 2017]. Their proposed architectures use pretrained word embeddings and represent topics as multivariate Gaussian distributions over the vector space of the word embeddings. Xun et al. improve the model of Das et al. by using the previously described CTM instead of the original LDA as base. The architecture by Xun et al., named Correlated Gaussian Topic Model (CGTM) outperforms LDA, CTM and the Gaussian LDA by Das et al. at topic coherence and document clustering. Lastly, some papers use word embeddings simply to precalculate word similarities or word correlations which are then used in the process of learning the model. These attempts can be demonstrated with two architectures that try to leverage the power of word embeddings to better represent short texts. In [Li et al., 2016] a Generalized Pólya Urn model (GPU) was used, which means, in simplified terms, that when a word is considered more likely to be in a topic during training, also all words that are similar to that word will gain a higher likelihood to belong to that topic. In a similar vein, Gao et al. [2019] use word embeddings to measure word distances. These distance are used to cluster short texts to create pseudo documents and to find global semantic correlations (two words that are close in the same pseudo document) and local semantic correlations (two words that appear locally in the same contexts). Only words that are globally and locally semantically correlated are considered correlated. These correlations are then taken into account when sampling from the topic distribition, which means word embeddings are indirectly influencing topic assignments. Gao et al. compare their model with LDA, PTM and GPU, which are all outperformed on the task of short text classification.[3]
Unfortunately, the architectures by Dieng et al. [2020], Xun et al. [2017] and Gao et al. [2019] have not yet been compared to each other.
Sadly most implementations of these algorithms are not public or are presented in a state that using them would take more time than this project would allow for. The algorithms and implementations I did test are presented in the next chapter.

[3]PTM beats the new model in one scenario, but performs much worse depending on a dataset, showing some instability that the new model does not suffer from.

3.3.4.5 Evaluation of Topic Models

When it comes to judging the quality of a topic model, the goal of the model has to be considered. If the model serves to prepare data for a downstream task, like classification in our case, other topics might be preferable than if the topics are immediately to be labelled by a human annotator.
In the first case, there is either the possibility of using the results of the downstream tasks such as automated classification, the evaluation of which will be discussed in the next section, or to use the log-likelihood of held-out documents. This value can, depending on the implementation, already be observed during training but when comparing topic models created by different implementations, it is important to check if the total log likelihood is computed in the same way [Wallach et al., 2009]. This measure is considered a good value to consider when the topic model is used for feature extraction as is the case in out task [Chang et al., 2009].
If the topic models are to be directly used by humans, it is important that they offer a high interpretability. One example how human judgement on the interpretability of topic models may be quantified is found in Chang et al. [2009] where the topics are scored by means of *word intrusion* and *topic intrusion* tasks. For word intrusion, the subject is presented with the five most probable words from a topic and one word with low probability in that topic, the intruder. In case of a good topic model, the subject should be able to identify most of the intruders. In the topic intrusion task, the subject is presented a document and must identify one out of four topics which does not fit the document. Again, in case of a good topic model, the human should identify the intruding topic most of the time. Of course human evaluation needs a lot of resources, especially if many topic models generated by different means are supposed to be evaluated, and is thus not applicable to this project.
A proposed metric which can be calculated based on the corpus is *topic coherence*. A number of different measurements exist to calculate topic coherence, which are based on the word probabilities in relation to the documents and parts of documents in a corpus [Röder et al., 2015]. In chapter 5, we will observe how coherence and log-likelihood correlate with the results of the classification tasks.

3.4 Classification

To know which classification algorithms work for our task, it is important to recognise the kind of multi-learning problem. In our case, each letter can be labelled with as many tags as needed, and each label is either present or not, we do not want

a probability for each label, but a binary True or False. This lets us define this problem as a so called *multi-label* classification.
Any classifier can be used in theory for multi-label classification, simply by training it once for each label in a binary manner and then combining the results of all classifiers, which is called a Binary Relevance Classifier. The disadvantage of this method is of course that any correlation between labels is lost, as each classifier is trained by itself. Another approach is to transform the task into a multi-class problem. This can be done by viewing each combination of labels as a class. Most classification algorithms support multi-class classification inherently. Of course this approach will not scale well if many labels are provided and the available training data is small, as some combinations of labels might only appear a few times or not at all. This is the case for our data, making this approach unsuitable to our task.
The Classifier Chain, proposed by Read et al. [2011], solves the problem of the assumed independence of labels when training a classifier for each label separately. In the Classifier Chain, one classifier is trained for each label, but each classifier that has already been trained passes on information to the next classifier. By doing so, interdependencies between labels can be used to improve the classifier performance. A problem of the Classifier Chain is that the results can vary significantly due to the order of the labels. To reduce this variance, Read et al. propose to use ensembles of Classifier Chains that vote on the final label assignments.
A number of classifiers are also inherently capable of multi-label classification, at least in their sklearn implementations, such as Decision Tree classifiers (e.g. [Dumont et al., 2009]), K-Nearest Neighbour classifiers (e.g. [Cunningham and Delany, 2021]) and Multi-Layer Perceptron classifiers (e.g. [Lenc and Král, 2016]).
In Kowsari et al. [2019] an overview is given on each of these algorithms (Multi-Layer Perceptrons are generalised as "Deep Learning"). In the following sections, I will summarise these findings and how they relate to our project and data.

3.4.1 Decision Trees

A decision tree outside of machine learning describes a tree-like structure where, after starting at the root, at each node a decision is taken with the final leaf that is reached being the outcome of the tree. When using decision trees for classification, which can also be done in a rule-based approach, the challenge is to decide which features are being taken into consideration and in what way at each node. With machine learning, this is done by maximising the information gain for each node.
Decision tree classification offers the advantages of fast training times and interpretable classification, as decision trees can be visualised. Disadvantages are that

they are prone to overfitting, meaning they will not generalise well on data outside the training set. Decision trees also tend to show a high variance in multiple runs, especially when even small changes occur in the training data. To offset this variance, Random Forests, which are ensembles of decision tress, were introduced. When using Random Forest, a set number of decision trees are trained which then vote on the classifier output. Random Forests retain the quality of fast training times, but are slower at prediction and lose the advantage of transparency, as only singular trees can be observed.
Because our data set is comparatively small and the prediction for the collection only has to be done once, training and prediction time are of low relevance. The possibility of observing the decision tree might prove useful though and will be investigated in the following chapter.

3.4.2 K-Nearest Neighbour

K-Nearest Neighbour classification, abbreviated as KNN, searches for each element to predict the K (a number set by the user) closest elements from the training set and uses their assigned classes to decide on the output.
The main downside is that the KNN algorithm is computationally very expensive, which again is not problematic to our project due to the small size of the training data. Additionally it can prove difficult to find the best value for K and the best algorithm to calculate the distances, which means for our project that multiple iterations of training would have to be performed to find the ideal settings.

3.4.3 Multi-Layer Perceptron

The Multi-Layer Perceptron (short MLP) is a type of feed-forward neural network which consists of at least three layers: The input layer, the output layer and the hidden layer in between. A MLP can also feature more than one hidden layer. When used in multi-label classification, the number of output nodes is equal to the number of labels. The output for each label must reach a set threshold so the sample is predicted to be assigned with that label. The assigned labels are then all respective output nodes that reach the threshold.
MLPs share the usual disadvantages of deep learning systems, in that they require large amounts of training data, are computationally expensive and offer no interpretability. But if enough training data is provided, deep learning systems can be expected to perform most other approaches. For our experiment, the small size of

training data might prove an obstacle, and lower performance must be expected.

3.4.4 Support Vector Machine

I will shortly describe Support Vector Machines (SVM) as well at this point, because they will serve as the Base Classifier to be used with the Binary Relevance and Classifier Chain methods of classification.
Imagine all training samples from two different classes to be mapped to a 2-dimensional plane, an SVM tries to find a line with which to separate classes the best as possible. For a higher dimensional space, SVMs use a x-dimensional hyperplane to separate the data points, where x is the dimensionality of the data points minus 1. Because this system would only allow for linear classification, a method known as the *kernel trick* is used to enable non-linear classification as well [Boser et al., 1992]. When predicting labels, the SVMs observe on which side of the hyperplane the sample is located and label it accordingly.
SVMs have proven to be very efficient classification algorithms. A major advantage is their robustness to overfitting, which is useful in case of our small data set. The obvious disadvantage is that SVMs are not inherently able to perform multi-label classification, so it is necessary to resort to the previously described methods to use SVMs for our task.

3.5 Evaluation Metrics

A number of evaluation metrics can be identified which will be useful to evaluate the output in the following chapters. Due to the task being a multi-label task, some of these metrics can only be applied to individual labels. This is the case for two common measures to judge the quality of a classification task: *Accuracy* and *F-Score*. Accuracy represents the fraction of samples which have been labelled correctly. In multi-label tasks, this measure can mislead, as it requires the all labels assigned to a sample to be correct so the sample will be counted as correctly labelled. In our case, with 17 labels, if even one is wrong, the whole sample will be counted as falsely classified. Thus accuracy will only be used when evaluating individual cases. F-Score is the harmonic mean of precision and recall. F-Score is also only applicable to individual labels. Accuracy and F-Score, and its underlying values of precision and recall, can be calculated for all labels by averaging the scores for the individual labels. If a data set is imbalanced, meaning some labels are far more prevalent than others, a weighted average can be used to draw a more accurate picture.

A very straightforward way to evaluate multi-label classification can be found in the metric of *Hamming loss*. Hamming loss is simply the fraction of all labels that have been correctly predicted. Unlike accuracy, this is done on a per label basis, not a per sample basis, so labelling one label incorrectly in a sample does not invalidate all the correct labels. The *Jaccard similarity coefficient score* measures the similarity between the true and the predicted labels of a sample in multi-label scenarios. Then the average of all similarities can be calculated.
In the following chapters, I use Hamming loss as a general value to score the experiments, and use recall, precision and f-score when observing the performance on individual labels.

4 Experimental Setup

4.1 Step Wise Grid Search

Due to the many factors that go into the final result, it is not possible to run all possible combinations of those factors in this project. Consider that even with only 3 different data sets, 3 different modes of preprocessing, 3 different feature extraction algorithms, 27 different runs would have to be executed. In reality, many more factors are relevant for each of these steps, and 16 different modes of preprocessing and 80 different feature extraction algorithms with different settings would be closer to what would actually be needed to be executed to get a full picture. To restrict the number of runs in a reasonable way, in the next chapter, different factors in the pipelines are investigated. For each of these factors, for example the choice of data set, I ran experiments where that factor is be the variable to be tested, while the others are be fixed, or at least narrowed down to a small range.
These fixed values have been determined by initial experiments and will be called the *default* settings. For each factor, the default value is named in the following sections (such as modern German for the data sets).
Of course this method might miss an unlikely combination of settings, such as a extremely minimum document frequency threshold with an extremely low number of topics in the topic model settings leading to better results than one would reasonably expect.

4.2 Data sets

As reported in chapter 2, the Bullinger collection features documents in either Early New High German or Latin, some of which contain regests in modern German, and a number of the Latin documents contain a modern German translation. Only Latin letters are available in sufficient numbers to serve as training data.
Five data sets can be derived from this collection:

Latin
All letters with majority Latin texts, 6613 in total. This is the largest single language corpus and thus offers a lot of documents to train the feature extraction algorithms on.

Modern German
This data set features the 2070 originally Latin documents, translated to modern German. Initial tests also concluded that adding the regests to those letters which contain a regest improved the results. This data set features the regest plus the translated text for each document, with no distinction made between the two. This data set is used as the default in the pipeline.

Latin Extended
This data set extends the original Latin texts with the regests in modern German wherever available. Thus this data set features 6613 documents as well. The idea behind this data set is to leverage the strength of the big size of the Latin data set together with the more standardised nature of the modern German regests.

Regests
This data set only features those letters with regests. The advantage of using this data set is that the trained model would be applicable to documents in Early New High German or Latin, as long as they contain a regest. This data set features 3104 documents.

Early New High German
This data set contains 1589 documents. As I do not have sufficient training data in the Bullinger collection, we will investigate with examples how topic modelling performs on this data. I will then speculate based on the examples and the achieved topic coherence how well classification would work if sufficient training data was available.

4.2.1 Using THBW data

All of the previously mentioned data sets (except modern German) can be expanded by adding the respective documents from the THBW collection. Because these letters are not exactly of the same genre and especially the regests are written very differently (See 2), per default, experiments are performed on the Bullinger collection alone. Section 5.7 will go into detail how adding THBW data can support the classification process.

4.3 Preprocessing

The choice of preprocessing methods can be crucial to the final outcomes as explained in chapter 3. I chose some general preprocessing because it showed success in the initial, unreported, experiments: Lowercasing, lemmatisation and removal of tokens with a length below 2 (Noise removal), which are usually punctuation.
Further I will explore how the removal of stop words influences experimental results. For this two different approaches are chosen to remove stop words: The first is a cautious method which only removes stop words that were annotated as such by the tagger (See section 4.3.1 below). The second is a more radical approach which filters out all words which are not tagged as either nouns, proper nouns, adjectives or verbs.
I also employ a filtering of words which only appear in a few number of documents, a measure named in Tang et al. [2014] as a common preprocessing step in topic modelling. I will evaluate how this influence the classification outcomes.
Default preprocessing employs the filtering of words by their Part-Of-Speech tags and sets a minimum document frequency of 25.

4.3.1 Implementation

The preprocessing of the Latin texts was done by employing the Classical Language Toolkit [Johnson et al., 2021], a natural language toolkit for Python which offers pretrained models for the Latin language. CLTK incorporates a number of tools from other platforms, such as Stanza [Qi et al., 2020], a tool developed by the Stanford University, for Part-Of-Speech annotation of Latin. It handles tokenisation well, leaving the abbreviations intact. I was not able to obtain any evaluations on the performance of the CLTK tools, but from manually checking some samples, the lemmatisation and annotation seems usable. Any self-reported results would not be transferable to our data in any way, as CLTK is trained on classical Latin texts from antiquity, which differ in genre and style from medieval texts.
For the regests and the translations in modern German, I used the spaCy framework[1], which also provides pretrained models for German Part-Of-Speech-tagging, Named Entity Recognition and more. spaCy achieves self-reported accuracies of around 97% in lemmatisation and part-of-speech-tagging on German news texts. I must expect the system to perform worse on my texts due to the difference in genre. Manual investigation of some samples has not shown any significant problems when using spaCy on my texts.

[1] https://spacy.io/

Example of an annotated token, uniform design for both Latin and German:

```
<token ne="" lemma="Zeit" string="Zeiten" pos="NOUN" stop="F" punct="F"/>
```

4.4 Calculation of Feature Vectors

The preprocessed data is then passed to the respective feature extraction algorithm. The following sections describe the chosen feature extraction algorithms and the used implementations.

4.4.1 TF-IDF

I implemented feature extraction by TF-IDF through the sklearn module[2]. Dimensionality reduction is then performed on the resulting vectors using latent semantic analysis, the predecessor of the pLSA method described in chapter 3, again using the implementation by sklearn. I experiment with different vector lengths. TF-IDF is one of the default methods of the testing pipeline, setting the numbers of target output dimensions to 50.

4.4.2 Word Vectors

To represent document representation via word vectors, I test Doc2Vec (See 3), using the implementation provided by gensim[3].Doc2Vec is not part of the default pipeline and will only serve to compare the final results to. I use Doc2Vec instead of the Fasttext document feature extraction, because the Fasttext feature extraction on document level is not publicly available and would require rebuilding it from scratch while Doc2Vec is.
During the work on this thesis, I decided to focus my experiments on TF-IDF and topic modelling, as those showed significantly better results than word embedding representation techniques. In the next chapter, I will report the scores achieved by Doc2Vec, but note that no exhaustive research was done in that direction.

[2] https://scikit-learn.org

[3] https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html

4.4.3 Topic Models

The basic LDA model and the Correlated Topic Model by Blei and Lafferty [2005] are used to demonstrate the ability of topic models for this task. There was no implementation for the improved CTM by He et al. [2017] available and due to the relatively small size of the Bullinger collection, training time was not too long, as long as the number of topics is not set too high. I investigate Pseudo-Document Topic Models as well, but only on the regest data set, which while still far longer than what PTM is usually used on, is at least shorter in average than the original texts.
I experimented with the algorithm proposed by Dieng et al. [2020] as well to incorporate a new method which uses word embeddings, but all results in initial runs did not improve the results over the baseline. While some runs managed to produce relatively coherent topics, but still worse than LDA and CTM, the resulting document-topic distributions did not lead to a classification quality better than the random baseline. It is unclear if this was due to a technical problem or if the topic distributions were still too bad to be used as features. In any case, the experiments based on this approach will not be further documented.
LDA and CTM are part of the default pipeline with the parameters of 50 topics, α prior set to 0.1, η prior set to 0.01 and number of iterations set to 5000.

4.4.3.1 Implementation

Tomotopy[4] offers an off-the-shelf framework to create topic models in python. It provides implementations for Latent Dirichlet Allocation, Pseudo-Document Topic Models, Correlated Topic Models and others.

[4] `https://bab2min.github.io/tomotopy/v0.12.2/en/`

4.5 Classification

4.5.1 Train/Test-Split

The annotated data was split into five folds, so each time, the model is validated on 20% of the data. I decided on a higher number of folds because the data set is comparatively small and I wanted to retain more training data for each run.

4.5.2 Classification Algorithm

Because the classification was very quick to perform on the small training data set, I was able to compare all the described classification methods from chapter 3. I again used the implementations provided by sklearn. To summarise the evaluated classifiers:

- RandomForest (Ensemble of Decision Trees)
- Multi-Layer-Perceptron
- K-Nearest Neighbour
- Binary Relevance using Support Vector Machine
- ClassifierChain using Support Vector Machine (In an ensemble of 10 chains with majority voting)

All classifiers are part of the default pipeline using their default parameters set in sklearn. I report the effects of fine-tuning the best performing classifiers in the next chapter as well.

5 Results

In this chapter, I will present the results to the experiments that were conducted with the aforementioned implementations. The first sections will discuss the effect that varying settings chosen for the preprocessing, feature extraction and classification have on the classification quality. Then the best combinations of those settings are presented and the results are discussed in detail supplemented by some examples. After that I will discuss how the THBW collection and the author metadata can be used to improve the classification quality. Finally, I will observe the quality of the classification when it is performed on the Early New High German texts by inspecting some examples.
The aggregated results of the classifications are reported as the Hamming loss, as it can be represented in a single number (the lower, the better) and is an easy way of comprehending if the model has improved or not. When investigating individual labels, Precision, Recall and F-Score may be used to offer a more extensive insight into where the classification performs well and where it does not.
For comparison of the results in the following chapters, I will give the random baseline of the Latin annotated Bullinger documents here: The best loss a classifier trained on randomised features achieves is 0.2369, which means that roughly a quarter of all label assignments are decided falsely.

5.1 Results On Different Data Sets

The first decision in the pipeline is which data set is used. Figure 8 shows a comparison of the default pipelines on the different data sets. There are multiple data points for each feature extraction algorithm as each point represents the output of one of the different classification algorithms. Note that lower scores are better, as Hamming loss is reported. The modern German data set significantly outperforms the other data sets with the extended Latin data set performing similarly at least when TF-IDF is used as the feature extraction algorithm. The quality of classification by only using regests relies heavily on the method of feature extraction chosen. While LDA performs much worse than on the extended Latin and the modern German

Figure 8: Comparison of data sets.

data set, TF-IDF does not suffer that much of a loss. This is probably due to the strong reduction of document length when word filtering is applied to the regests. The median length of regests drops down to 7 words per document in this case, which is too short for common LDA algorithms (Compare section 3.3.4.3). This indicates that CTM and TF-IDF are better at working with these shorter documents. Not shown in the figure is which data point refers to which classifier, but it should be noted that the best results were achieved for all sets and all feature extraction algorithms shown here by the One-Vs-Rest and the Classifier Chain classifier (the two data points at the bottom of each scatterplot), which is a consistent trend.
I included PTM as well for the regest-only data set, but as is observable in Figure 8, the pseudo-document topic model does not outperform CTM and TF-IDF. It does offer a considerable improvement over LDA though, which it is based on.
These results indicate that the modern German data set is most suitable to train the feature extraction on. The errors in the translations do not matter too much, because sentence structure and grammatical correctness are not a factor for the feature extraction algorithms which all work with lemmatised data and bag-of-word approaches which do not take sentence structure into account. Modern German has an advantage over the extended Latin data set in that it only contains one language, which helps creating more consistent topics in case of topic modelling and a much

smaller vocabulary in case of TF-IDF.

5.2 Effects Of Preprocessing Choices

5.2.1 Stop Word Filtering

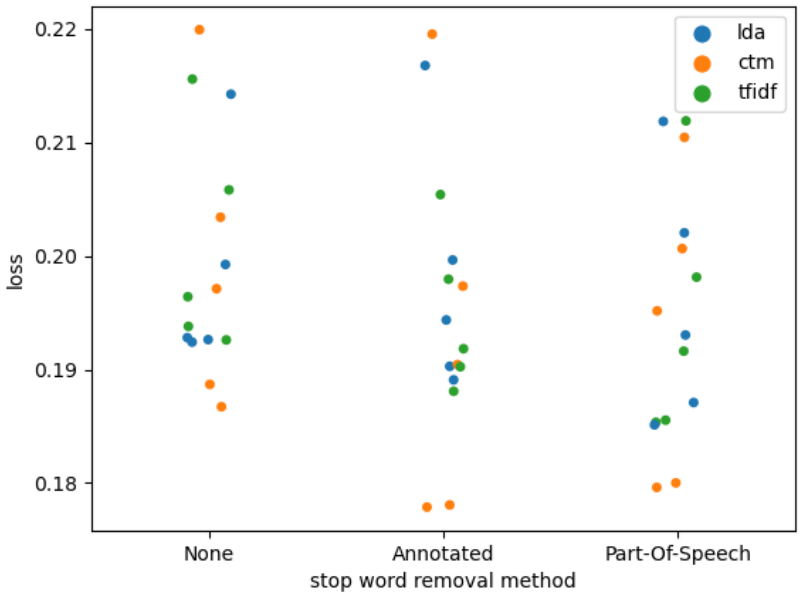

Figure 9: Comparison of effects of method of stop word filtering.

In Figure 9, three different methods of handling stop words are presented. The two methods are, as described in the previous chapter, filtering of the stop words marked as such by the tagger or filtering stop words by removing all words that are not tagged as either nouns, verbs, adjectives or proper nouns. Again, each data point represents the result by the shown feature extraction by a classifier, the performance of classifiers varies strongly. As expected, no stop word removal performs the worst. CTM performs much better when only words annotated by a tagger are removed while LDA and TF-IDF show a slightly better performance on the POS-filtered data set.
In terms of vocabulary reduction, while the modern German data set has a vocabulary size of 37'923 after lemmatisation but before preprocessing, it is reduced to 3'942 by the low frequency word filtering (see next section) and further reduced to 3'640 with the annotation method or to 3'197 unique words when the POS-filter method is employed.

This means that while LDA and TF-IDF work better when the data set is cleaned more thoroughly of low-information words, CTM suffers from the removal of some words that might bring some information with them.

5.2.2 Low Frequency Word Filtering

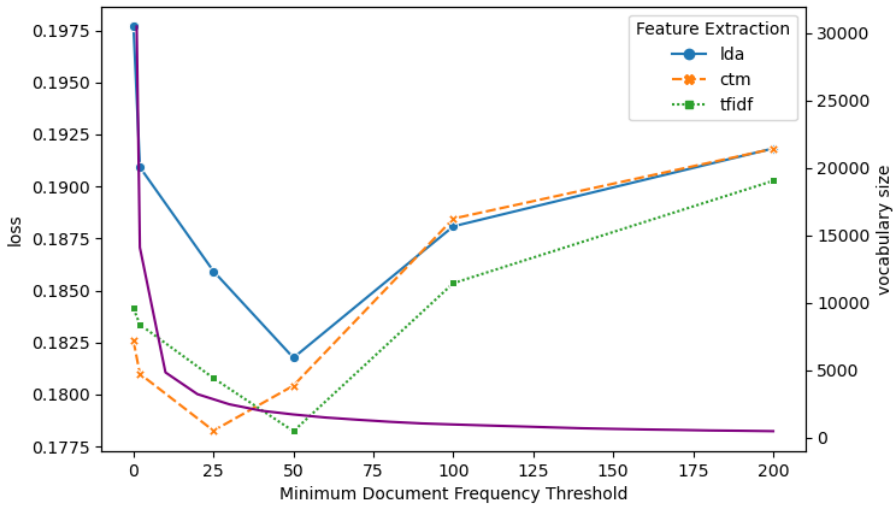

Figure 10: Comparison of effects of low frequency word filtering. The purple curve represents the remaining vocabulary size at each threshold.

Figure 10 offers insight on how the filtering of words which only occur in few documents influences results. Only the best classification results are reported in the figure, which are either the One-Vs-Rest or the Classifier Chain classifiers. A clear trend is visible that filtering out words with low frequency improves the quality of the feature extraction. Setting the threshold too high may cause adverse effects, as observable in the figure. Note that CTM performs better with less word filtering than LDA and TF-IDF. This indicates that CTM is able to make better use of low frequency words than the other two systems and is more dependent on low frequency words than the other systems.
Considering the results in the previous section, it can be stated that CTM does not only need less preprocessing than the other feature extraction algorithms to work well, but also that preprocessing has to be applied with caution as the removal of too many words may reduce classification quality. In general I can observe here that preprocessing modes should be chosen individually for each feature extraction method, because there is no mode that fits all feature extraction methods equally well.

5.3 Effects Of Dimensionality Reduction in TF-IDF

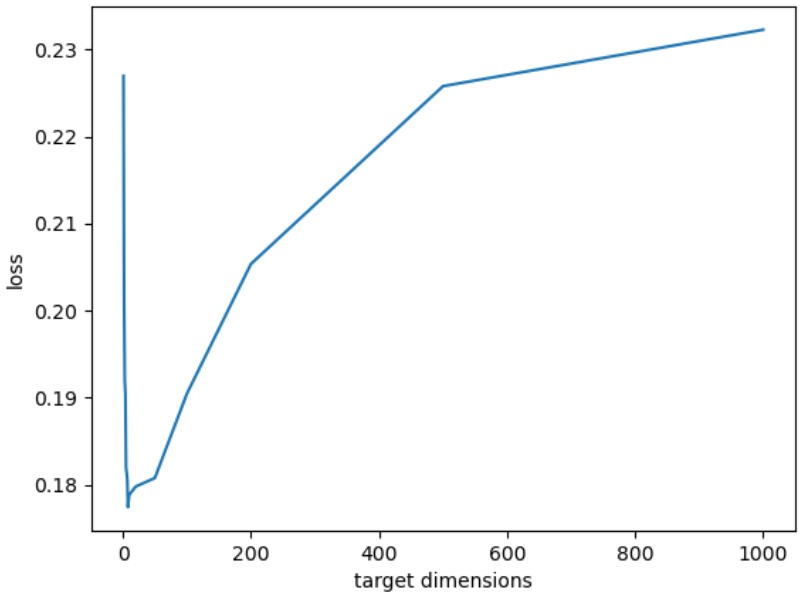

Figure 11: Dimensionality reduction effects on TF-IDF.

As explained in chapter 3, dimensionality reduction is an important step when using TF-IDF to reduce the long vectors to their most important features. Figure 11 illustrates that the classification works especially well when using few target dimensions. Best results are achieved when using ten target dimensions, but very similar results are also achieved when setting up to 50 target dimensions.

5.4 Effects Of Topic Modelling Parameters

In this section I will compare how different parameter settings for topic models influence the results. For each of the parameters k (number of topics), α (document-topic Dirichlet distribution prior for LDA / smoothing value for CTM), β (topic-word Dirichlet distribution prior) are investigated, as well as the change with the number of training iterations.
The effect of k is visualised in Figure 12. I tested the numbers 10, 20, 30, 40 and 50 for k. The figure represents all results by different iterations from 1000 to 10000, in steps of 1000 (so 10 times 5 data points for each k, for 10 different iteration settings and 5 different classification algorithms). For LDA, the best performances are achieved with 40 topics, with models of 50 topics performing significantly worse. For

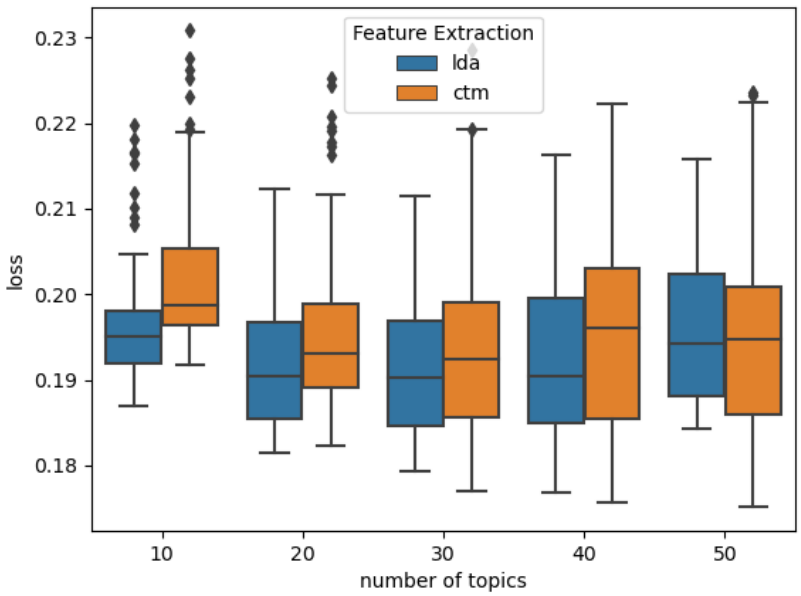

Figure 12: Comparison of effects of different topic numbers.

CTM while no trend can be observed in the average scores, best results are achieved with a high number of topics. The large variance for CTM is likely due to it not performing well if few training iterations are done, but this will investigated further below. The best performing models in this figure are $k = 50$ with 9000 iterations for CTM and $k = 40$ with 7000 iterations for LDA. It can also be observed that a k of below 30 is unable to achieve comparable results to a higher k. Especially with CTM though, a high k brings considerably longer training times, which is the reason the experiments here were capped at 50 topics, so enough experiments could be performed.

The same kind of comparison is visible in Figure 13 for the α and β parameters. My assumption in chapter 3 was that due to the strong mix of topics in the Bullinger documents, a higher α would produce better results. For α, there is no clear trend for either of the systems, with the lowest setting as well as the highest setting tested achieving good results for LDA. CTM shows the best result if α is set to the default of 0.1. Neither is there a clear trend visible with the different β settings, but CTM and LDA both achieve the best performances when given an β of 0.05, five times higher than the default of 0.01.

The success of a high α for LDA in this experiment confirms the point made in [Tang et al., 2014], which says that if a document only contains few topics, α should

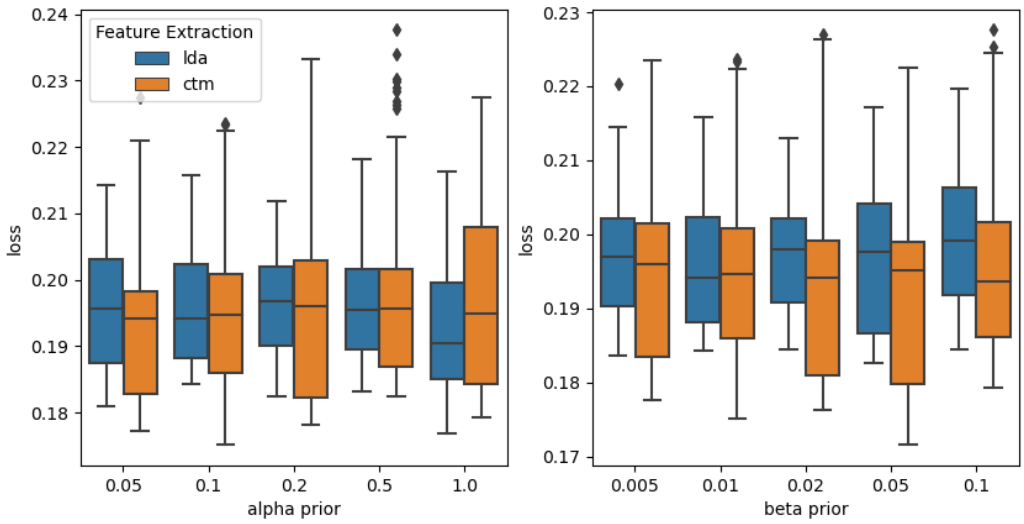

Figure 13: Comparison of effects of different α and β settings.

be set small. This does not seem to be true for CTM, where α has a different role in the model algorithm, being only a smoothing value, not a distribution prior. The documents in this collection contain a strong mix of topics, with each letter potentially being about world news, education and personal affairs, compared to e.g. news articles which are far more homogeneous. The relatively successful large β indicates that the Bullinger collection produces relatively similar and not well defined topics, according to Tang et al., which is a disadvantage in general when trying to learn topics. This would explain our relatively small difference in performance between topic modelling and TF-IDF.

To compare iterations I included all previous experimental runs and present them in Figure 14 as a lineplot with vertical lines to show the standard deviation. For CTM, a slight trend to better models with more iterations is visible. For LDA, no such trend is visible. In general, the standard deviation is extremely high, as visible in the figure. To demonstrate the variation from run to run, I included the results for three runs for each LDA and CTM at each iteration step, for each the best classification result is reported (usually provided by the Classifier Chain), represented as dots in the figure. The inherent randomness of topic models makes it very difficult to find a trend and one could speculate that the best method to find a good topic model would be to just run the experiment with reasonable parameters for a large number of times and then pick the model that performs best downstream. Not forgetting that there is also randomness inherent to most classifiers, which increases variation

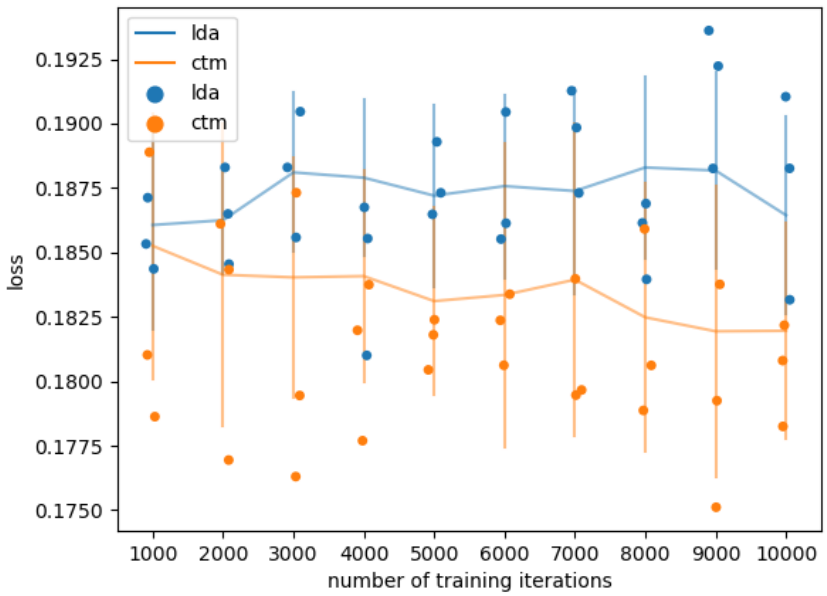

Figure 14: Comparison of effects of different number of training iterations.

in the results as well.
Because running a large number of experiments for each setting is outside the possibilities of this project, especially for CTM, for the remainder of this chapter I will assume that the values gained by the individual runs are representative of the general quality of these parameter settings.

5.4.1 Topic Coherence and Log-Likelihood

A pre-selection of topic models by observing their coherence score or log-likelihood would make it possible to omit the randomness of the classifiers. In this section, I will test if there exists a correlation between either of those values and the best classification score.
Figure 15 presents the correlations between coherence, log-likelihood and the final score. I included the best classifier results for all models generated with CTM from the previous section, restricted to models with 50 topics. As expected, a strong correlation between coherence and log-likelihood exists (Pearson correlation coefficient of -0.863). A weaker correlation is visible between log-likelihood and the final scores, as well as coherence and the final scores. Note the negative correlation between coherence and scores (Pearson correlation coefficient of -0.32), supporting the notion

that topic coherence, while being a good indicator for human interpretability of the the topics, is not necessarily a good indicator if the model performs well on downstream tasks like automated classification. In this case, a higher topic coherence even notes lower scores. The positive correlation of log-likelihood and final scores is what would be expected (Pearson correlation coefficient of 0.28). While a weak correlation can thus be observed, it is important to note that many models with higher log-likelihood perform worse than a lot of models with low log-likelihood. Thus I would not use the value of either of these metrics as a pre-selection method unless the training of the classifiers is expected to be so time-expensive that only very few models could be used to train on. This is not the case in this project, as the creation of the topic models itself is the most time-expensive task.
Table 7 demonstrates the advantage of a high coherence when it comes to hu-

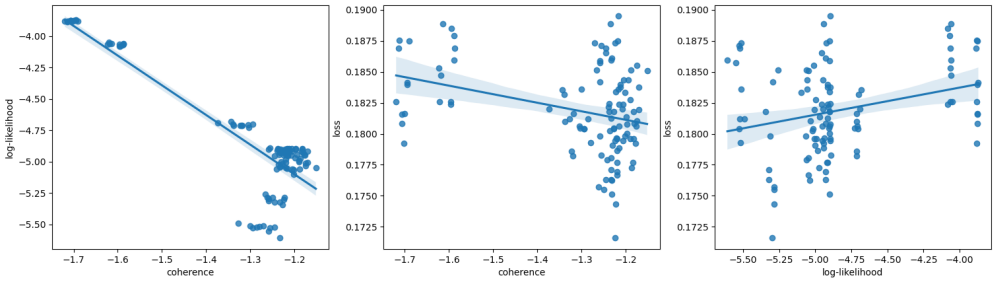

Figure 15: Correlation of coherence, log-likelihood and scores for CTM runs.

man interpretability. Included is a model with high coherence (LDA, $k = 50, \alpha = 0.1, \beta = 0.01, iter = 6000, coherence = -1.110$) and one with low coherence (CTM, $k = 50, \alpha = 1, \beta = 0.01, iter = 2000, coherence = -1.721$). For each model, I present two topics which include the three words soldier, war and frenchman.The high coherence model contains two well separated topics, one that includes words about general warfare such as *Heer* (Army), *Soldat* (Soldier) or *Lager* (Camp), and another which contains person names, country names and words derived from country names, such as *Franzose* (Frenchman), *Türke* (Turk) and *Heinrich*. The low coherence model does not contain such topics, but instead mixes both of them. The first topic contains general warfare concepts as well, but also the words frenchman and turk, while the second contains also general warfare terms but also words about how news about warfare are delivered such as *berichten* (report) and *erzählen* (tell). If one was to label the topics by hand, the high coherence topic model would likely be more suitable to do so. Interestingly both perform almost exactly the same when used as feature vectors, producing a hamming loss of 0.186.

High Coherence 1	kaiser heer stadt soldat truppe ziehen spanier krieg lager reiter
High Coherence 2	könig türke franzose franzen frankreich französisch ferdinand kaiser heinrich krieg
Low Coherence 1	kaiser könig karl heer soldat truppe franzen franzose türke frankreich
Low Coherence 2	fürst berichten krieg papst gesandte deutschland eidgenosse ziehen deutsch erzählen

Table 7: Topic top words containing the word *Soldat* (soldier), *Krieg* (war) and *Franzose* (frenchman) in models with high resp. low coherence.

5.5 Classifier Comparison

Surprisingly, none of the inherently multi-label capable classification methods perform comparable to the expanded SVM. Figure 16 visualises the distribution of scores on a broad number of vectors produced from LDA, CTM and TF-IDF.
One-Vs-Rest (OVR) and Classifier Chains (CLFC) based on Support Vector Ma-

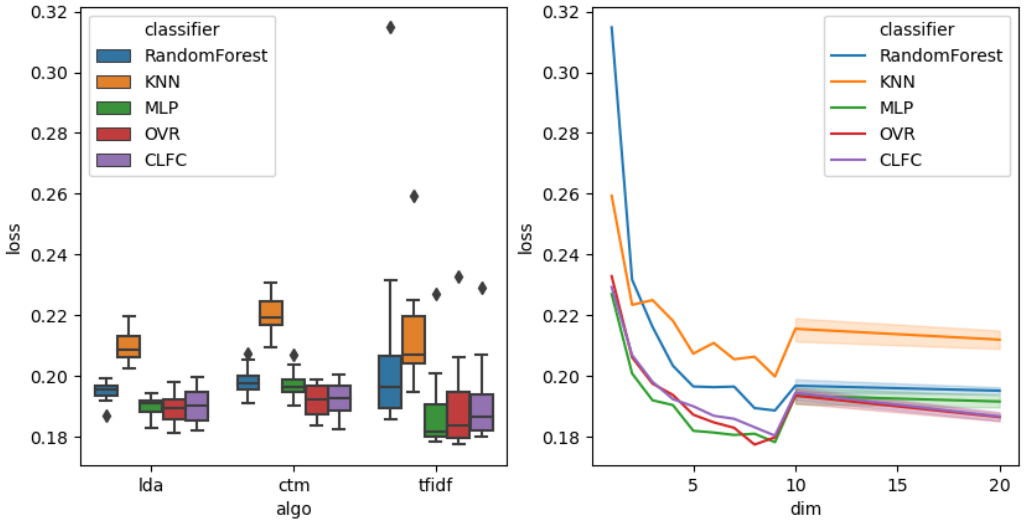

Figure 16: Comparison of classification scores by each classifier distinguished by the feature extraction method.

chines clearly outmatch the other systems on LDA and CTM trained vectors. The only exception to this is in the very low dimensional space (< 20 dimensions), where the Multi Layer Perceptron classifier performs to a similar or even slightly better quality. This indicates that MLP is more comfortable with low dimensional feature vectors. Note that the seemingly better performance of MLP on TF-IDF in the left figure is because proportionally more TF-IDF experiments were run with low

dimensional vectors than for LDA and CTM. MLP still performs significantly worse than OVR-SVM and CLFC-SVM when used on models with higher dimensional vectors (Example: TF-IDF with dimensionality reduction to 50: MLP 0.193 loss, OVR-SVM 0.1807 loss).
Fine-tuning was difficult to do due to the unavailability of fine-tuning tools for multi-label tasks in sklearn. I still tested a number of options for each classifier, but no significant change was visible.
A surprise was the ineffectiveness of the Classifier Chain method, which did not lead to a significant improvement over the One-Vs-Rest method. Both score an extremely similar average score of 0.18465 respectively 0.18470. I interpret this as the relationships between labels being too vague to support the classification process. The other methods average score were far behind that with RandomForest scoring 0.1962, KNN 0.2130 and MLP 0.1947.

5.6 Best Performing Pipelines

Method	Min Doc Freq	Parameters	Classifier	Hamming Loss	F-Score
TF-IDF	50	10 dimensions	MLP	0.1754	0.5237
Doc2Vec	25	50 dimensions, 40 epochs	OVR-SVM	0.1961	0.4575
CTM	25	$k = 50, \alpha = 0.1, \beta = 0.05$	CLFC-SVM	0.1716	0.5371

Table 8: Top scoring models for different feature extraction methods.

Following the experiments reported in the previous sections, a grid search was applied for all parameter combinations which seemed reasonable that had not yet been tried. For TF-IDF, this was combining the strong dimensionality reduction (dimension size 8 to 12) with different minimum document frequency thresholds (from 2 to 50), which indeed managed to produce the best result for TF-IDF, reported in table 8. For CTM, larger topic numbers were tested as well as more iterations and different β values. This produced the result reported in the table, but note that this is likely an outlier, later tries with the same parameters could not reproduce this result. Doc2Vec was also tried with a number of different parameters, but not as exhaustive as the other methods and produced considerably worse results than those. Note that the F-Score in the table does take the label imbalance into account by using the weighted average. Exact scores produced by the topic model are reported in table 10 which can be found in appendix A.
For convenience, the scores are also visualised in Figure 17. Some labels, like *Education* and *Conflicts*, are only applied with caution, showing a high precision, but

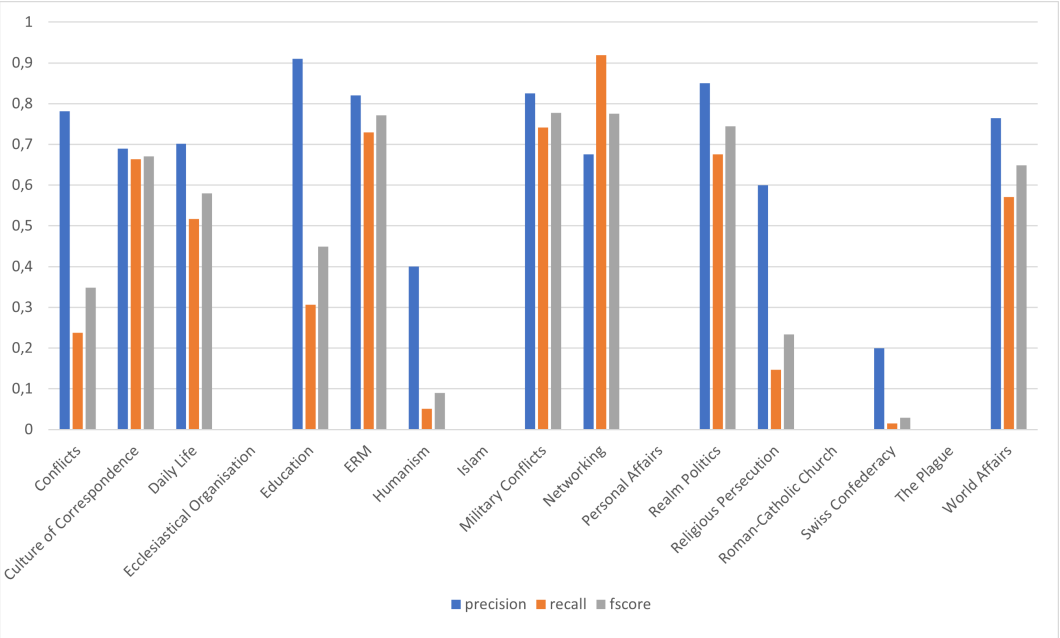

Figure 17: Precision, recall and f-score for each label assigned by the best pipeline.

only a low recall. The labels which were not assigned at all, like *Ecclesiastical Organisation*, are coincidentally also all those tags with the lowest occurrence in the collection. It seems that due to the fact that these labels only appeared so rarely, the system gained the best results by not labelling them at all. It is to note that here, even though *The Plague* and *Islam* would both be identifiable by one keyword each ("Koran" for *Islam*, "Pest" for *The Plague*), this information could not be used by the classifier. This is an obvious disadvantage of document representation via topic modelling. In this case, a simple keyword-search or a basic rule-based algorithm to annotate the documents would likely be more efficient.
The labels which occur more often in the training data such as *Evangelical-Reformatory Movement* and *Military Conflicts* attain relatively good results. I will observe how these results translate for individual samples in the next section.

5.6.1 Error Analysis

In this section I will investigate some of the errors the best pipelines make, but also what they are doing correctly. The first example is a letter from the Bullinger collection, written by Bullinger to Myconius.[1] Bullinger first informs Myconius, who lives

[1]HBBW 2304.

in Basel, about the Ottoman war campaign, then goes on to talk about some treaty between the duchy of Milano and the Swiss Confederacy. Finally he names the messengers who will tell more about these subjects and sends his greetings to Johannes Gast, who also remains in Basel. The talk about the messengers and greetings are kept relatively short, while the news about the war and Milano make up most of the content. I annotated the letter as *Military Conflicts* and *World Affairs* due to the talk about the Ottoman campaign and as *Swiss Confederacy* due to the talk about the treaty of Milano being connected to the Confederacy. The topic modelling pipeline classified the letter similarly, but added *Culture Of Correspondence* and did not assign *Swiss Confederacy*.

This letter will serve as an example why the system rarely assigns *Swiss Confederacy*. First, "Eidgenossenschaft" (Confederacy) and "eidgenössisch" (federal) occur once each in the filtered text. While other words related to the Confederacy occur as well, like "Zürich" and "Basel", these words are frequent in all letters, due to them being part of the date and address parts. In this letter, the most important words like "Eidgenossenschaft" are also not part of the text, but only appear in the regest (text in brackets is supplemented by the editor and not contained in the original text): *"Dies geht aus der Abschrift des Briefes hervor, die [d'Avalos] den [eidgenössischen] Gesandten übergab. Ohne diese Abschrift hätten die Gesandten das [für die Eidgenossenschaft bestimmte] versiegelte Schreiben nicht akzeptiert."* This again demonstrates the low frequency of these relevant terms for *Swiss Confederacy*. If I inspect the topics in the topic model, there is no topic that refers to the Swiss Confederacy specifically, unlike the labels *Military Conflicts* and *World Affairs* which are very well represented in multiple topics. For example, the document-topic distribution shows the topic 43 to be the most dominant, which includes the top words "könig papst franzose frankreich franzen ferdinand französisch gesandte england deutschland", and the topic 26 as the second most dominant topic with the top words "türke bund krieg stadt fürst türkisch hören bündnis versuchen feind". "Eidgenosse" is contained in topic 38, which is also assigned with a high probability to this document, but is more about warfare than about the Swiss Confederacy specifically, containing the top words "heer stadt soldat krieg eidgenosse helvetier kaiser truppe spanier ziehen". This topic does not seem to be correlating enough with the documents annotated with *Swiss Confederacy* to signal the classifier that this document belongs to that label. This highlights the importance of clearly defined topics.

Now to the incorrectly assigned *Culture Of Correspondence*. First, keep in mind, this is a label which appears a lot in the collection, more than in half of all letters. So the classifier, even without strong features, is likely to assign this label to any letter. In the topic model, *Culture Of Correspondence* is visible in multiple topics,

for example in topic 33, which contains the top words "schreiben brief antworten antwort überbringen". Topic 33 is also very dominant in the document-topic distribution, being assigned the fifth highest probability for that document (out of 50 topics). I wanted to test what influence single words have on the topic distribution. I removed all occurrences of "brief" and "überbringen" from the letter. These words appear in the letter in the context of the treaty with Milano (*"Der Brief [von Alfonso d'Avalos], der den aus Mailand zurückkehrenden Gesandten anvertraut wurde, [...]"*) This caused topic 33 to drop from the fifth most important topic to the eighth, reducing the probability of the topic being part of the document by half. This indicates that already single words can have a large impact on the topic distribution. Assuming the classification process uses the value assigned to topic 33 as an indicator that a document should be labelled *Culture Of Correspondence*, this demonstrates how a document like this would be labelled incorrectly.
In the next example, I will investigate the low recall of the *Education* label. Note that this label is assigned to relatively few letters with 66 annotations in total, which is similar to *Humanism* with 56 annotations, which shows a much lower recall, and *World Affairs* with 75 annotations, which shows a much higher recall. The interesting part about *Education* is that it has the best precision out of all tags. I will look at a letter by Rudolf Gwalther for this example.[2]
The correct labels feature not only *Education*, but also *Daily Life*, *World Affairs*, *Military Conflicts* and *Networking*. While all other labels are assigned when using the topic model pipeline, the *Education* label is missing. Gwalther tells Bullinger about his studies and informs him that he changed to a school in Morges. The text features signal words like "Schule" (school), "Kollegium" (college), "Student" (student) and "Studiengenossen" (Study companions). When manually inspecting the topics, topic 40 fits best for the label *Education*, containing the words "vater leben sohn jahr lehrer jung gut studie schule studium". This topic is also very domininant with the fourth highest probability for this letter. To compare these results I review the topic distributions by three documents that were correctly labelled as *Education*. These three documents had very similar probability values to the document that was not labelled as *Education*. I can only speculate why these documents were successfully labelled, while the inspected one was not. The classification obviously takes more than one topic into account when deciding on the label and due to the low amount of training data with a strong mix of subjects per document, the classifier has trouble trying to make the connection between single topics and labels. There could also be letters with a high probability for topic 40 which I did not label as *Education*, which make it more difficult to find this connection.

[2]HBBW 1298.

From these inspections, I learned that there are some very strong correlations between labels and topics which, while easy to identify by hand, are difficult to spot for the classifier. This brings up the question if this fully automated system could best be used to identify the most suitable topics for a human researcher who could then use these topics and their distributions to build a rule-based system incorporating this knowledge. A machine learning system with hand-crafted features would also be a possibility, passing only the probabilities for those topics to the classifier per label, that are deemed by the researcher most promising to identify that label.

5.7 Including THBW

The documents provided by the "Theologenbriefwechsel im Südwesten des Reichs in der Frühen Neuzeit (1550-1620)" will demonstrate if the supplication of additional data that is similar but not from the same source can help with the task. I will also test how the pipelines that performed well on the Bullinger collection perform on the THBW collection with its original tags as well as the converted tags. Note that there are no translated texts available for THBW, so the experiments are conducted with the extended Latin data sets (Regests + Latin text).

5.7.1 Annotating THBW

In a first experiment, I test how usable the tags of the THBW collection are. The random baseline is at a loss of 0.1340, which is reasonable as the numbers of assigned labels per document is much lower for THBW than for the Bullinger collection. The best performing pipeline is feature extraction via TF-IDF and classification by One-VS-Rest-SVM, which achieves a loss of 0.1144. This system only achieves an average recall of 22.06%, but a precision of 52.89%. The problem of the label imbalance, similar to what could be observed in the Bullinger collection, is also visible in the THBW collection, but the less frequent labels suffer even more from it. The only label with an acceptable annotation quality is *Evangelical-Reformatory Movement*, which is by far the most frequent label, with a precision of 78.08% and a recall of 69.15%. Even the second and third-most frequent labels *Daily Life* and *Education* only achieve a recall of 10.37% and 16.45%. This indicates that documents which are annotated as one of those two labels can not be distinguished well from documents which are not annotated with those labels. This is probably due to the way the annotations were created by converting the original labels to the Bullinger labels

(See section 2.2.2.1).
An example of an error counted like this can be found in a letter by Petrus Paciens, which the system annotated with *Education*.[3] In this letter, Petrus writes among other things about a theologian by the name of Edo Hilderich who arrived in Heidelberg and is to receive a doctoral degree soon, as well as who will supervise his promotion. The document is annotated with "Doktorpromotion" (doctoral degree). If I had annotated this letter by hand with the Bullinger label set, I would have annotated it as *Education* as well. But the system described in section 2.2.2.1 did not convert "Doktorpromotion", because it only occurs in eight documents, so it was not converted manually, and is not assigned to any parent label in the label hierarchy. Thus the label assignment is counted as a False Positive instead of a True Positive. This does not only make the score look worse than it actually is, but introduces erroneous information to the training data. The repercussions of having these errors in the data will be observable in the next section.
I will observe as well what happens when no conversion of the tags is performed. Instead, I use the original annotations, but limit the label set to those tags which appear at least 50 times in the collection. For those annotations that do not meet those thresholds, I look up the parent labels, if any exist, and include that one if it meets the threshold. This results in a label set of 43 different labels, with the most common labels being "Abendmahlslehre" (Communion Doctrine), "Konkordie" (Concord) and "Confessio Augustana" (Augsburg Confession), all of which would have been converted to *Evangelical-Reformatory Movement* in the Bullinger label set. The baseline of this experiment is at a loss of 0.05391. The best result is produced by the same pipeline as above and reduces the loss to 0.05121. 29 labels were not assigned at all. Considering these labels sometimes provide more training samples than the labels contained in the Bullinger collection, the problem is in the lacking distinctness of those labels. One label performs extraordinarily well "Augsburger Reichstag" (Imperial Diet of Augsburg) with a precision of 90% and a recall of 61.67%, even though it had less or around the same amount of occurrences compared to those labels which were not assigned at all. The reason for the success in labelling "Augsburger Reichstag" can be found in the mixture of labels in its documents. While all correctly classified documents contain only one or two labels, half of the False Negatives contain more than two labels. This also means that ca. 75% of all documents labelled with "Augsburger Reichstag" only contain one or two labels. Compare this to the label "Krieg" (War), not assigned once although it occurs twice as often in the data, only a third of all documents contain one or two labels. This serves as a good example how monothematic documents are not

[3]THBW 22244.

only easier to classify than documents with strong mixes, but also make for better training data.
The weighted average recall was at 7.26% with a precision of 23.29%. These scores are even lower than those achieved by the converted labels, reported above. This demonstrates how a smaller label set with more distinct labels is easier to classify. The methods described in this work are thus not applicable to assign fine-grained labels like the THBW collection contains. I would better look for key phrase extraction methods for this use case.

5.7.2 Adding Data

In this second experiment, I observe what happens when more training data is available. In a first step, I test what happens when the THBW documents are added to the feature extraction, but their annotation is not used. The default CTM gets a loss of 0.1842 without the additional documents while the CTM trained on the combined collections achieves a loss of 0.1835. For TF-IDF, I tested a variation of settings for dimensional reduction and word filtering to adjust for the changed size of the data set. The best pipeline achieves a score of 0.1835, compared to 0.1853 when only the Bullinger collection is used. Note that these scores are lower than the pipelines presented in the previous section because I use the extended Latin data set instead of the modern German data set. While the combined data set performed a little bit better than only the Bullinger data set, the differences are small and can be considered inside the expected variance from the topic model and the classification algorithms. But these results do confirm that adding the data does not make the feature extraction considerably worse.
In a next step, I make use of the annotations provided by the THBW collection, by adding those annotated document as additional training data. Note that I still only evaluate on the Bullinger collection. As observed in the previous subsection, the THBW annotations do not perform very well as training data. I will still test on this imperfect data as experiments in Machine Translation have shown that using imperfect data can help when not enough training data is available [Sennrich et al., 2016]. An important factor when using data which is expected to contain errors is how much data is added to the original training data. I conducted a number of experiments with different amounts of THBW documents added to the Bullinger training data. The documents that were added were chosen randomly. The results of the different trials can be observed in table 9.
While adding all of the data does worsen the performance of the pipelines, adding only part of it shows a significant improvement. When comparing the results of

Additional Documents	Loss	Feature Extraction
All (767)	0.1848	TF-IDF
300	0.1810	TF-IDF
150	0.1809	CTM
75	0.1804	CTM
25	0.1808	CTM
None (0)	0.1835	CTM

Table 9: Best Scores Achieved With Additional Training Data.

the original data set with those of the supplemented data set, significant changes are observable for some labels. For example, the precision for *Education* increases to 93.33% from 35.07% while the recall is lowered from 49.28% to 33.64%. In THBW, *Education* is the second-most assigned label, which is why I would have expected an increase of assignments. Possibly, the THBW training data includes some strong indicators for letters that should be labelled as *Education*, which leads to the system narrowing down what it considers to be *Education* and only assigning it to documents that contain those strong indicators.
The experiments of this subsection do not only prove that adding training data from a similar collection supports the classification process, they also prove that this is true even if the training data is partly labelled incorrectly. It also shows how more training data in general would be able to improve the system. Note that the best scores reported here are still worse than those resulting from the modern German data set, so these pipelines, even without additional training data should still be regarded as the best way of labelling the documents.

5.8 Using Author Information

As proposed in chapter 2, the information about the authors might provide information which can be used to support the classification. In a first step, I explore the connection between document and authors by using the authors as labels to the classification. As one document can have multiple authors, this is formulated as a multi-label task. I limit myself to only those authors who wrote at least 50 letters in the extended Latin data set, which resulted in 21 authors. I use the Latin data set here as the translated texts might lose information about individual writing styles. The result when using the same settings as the best topic model reported in section

5.6 is promising. While the random baseline shows a Hamming loss of 0.029, when using the topic model feature extraction, a Hamming loss of 0.0153 can be achieved.[4] Bullinger himself is correctly assigned to 76.04% of his letters and 88.56% of the predicted labels were correct. Because I suspected the names inside the letters to help during classification, I trained the topic model once on the data set with all named entities replaced by placeholders referring to their named entity type like "PER" for persons. Surprisingly, this yielded even better results with a Hamming loss of 0.0146. This difference can be explained due to the variance inherent to the topic models.
These results are not reflected when using the author data as additional features for classification. Again, using only authors who wrote at least 50 letters, the Hamming loss worsens from 0.1842 to 0.1862. The correlation between authors and tags and their relationship to the other features is not strong enough to improve the system.

5.9 Test on Early New High German

In this section, I will investigate how well the best pipeline from above works on the Early New High German texts. Because there is barely any training data available for German texts in the Bullinger collection, three data sets were tested: One using all regests of the Bullinger collection and one using the the German THBW documents as training data. A third data set combines the approaches by using both regests and German data. The scores in section 5.1 give an indicator what performance can be expected overall when only using regests. Note that when using the Bullinger data as training data, there exists a possible problem of representativeness, as documents written in German are likely to have a different label frequency distribution than the Latin documents. For example, a letter in German is much more likely to not have been written by a theologian, so it is less likely that themes from the *Evangelical-Reformatory Movement* label would be discussed. In this section I want to provide some examples by showing how three different letters were labelled by the data sets described above. The examples were randomly sampled, but I discarded some of the sampled examples if they were not interesting. The system used to assign labels to the examples was the best pipeline using the Correlated Topic Model feature extraction, but word filtering reduced to a minimum document frequency of 10 words (instead of 25), to account for the smaller data sets.

[4]Note that these numbers are so low because most letters only have one author or even no author at all, because they were filtered out because they wrote too few letters, so most instances of author-letter combinations are 0.

The first letter was written by Ambrosius Blarer to Bullinger.[5] Blarer tells Bullinger about an incidence in which some Swiss men met soldiers of the Emperor at Kempten and beat them after having been insulted. Further, he mentions that he heard that Swiss mercenaries are said to be joining the German troops and wants to know if it's true. The regest-only data set labels this letter with *Culture Of Correspondence*, *Realm Politics*, *Networking* and *Military Conflicts*. The models trained on the other two data sets which include the original German texts, do not assign any topics at all. The labels *Realm Politics* and *Military Conflicts* are debatable for this letter. While soldiers are part of the incident and it does happen in the context of a greater military conflict, the Schmalkaldic War, no direct military or political action is mentioned. On the other hand, the rumor about the Swiss mercenaries joining German troops could be labelled as *Military Conflicts*. I would not find this letter to be out of place when exploring these labels, so I count them as correct labels. This example demonstrates how difficult it can be even for a human to decide on a label. The *Culture of Correspondence* and *Networking* labels are assigned to most letters, so it is not suprising that this letters is labelled as such. In this case the document contains nothing which would refer to these topics specifically, so these labels are assigned incorrectly. The label *Swiss Confederacy* does only fit this letter partially, as it does mention a decision by the convention of the Confederacy regarding the mercenaries but no inner-Swiss politics are discussed. Thus I see it as correct to not label the document in this way. In total, two of the four assigned labels were correct and the assigned labels *Military Conflicts* and *Realm Politics* offer a quick insight what the letter is about.

The second letter was written by Bullinger and is addressed to Oswald Myconius.[6] In the first paragraph of the letter, which is actually written in Latin, Bullinger thanks Myconius for something which is not mentioned in the text. The regest does supplement the information that he shows his gratitude for the goodwill shown to a delegation from Zurich at a meeting in Basel. He then goes on to report that the First Helvetic Confession, a document drawn up and signed by protestant theologians from the Swiss Cantons, among them Bullinger, which aimed to approach Luther in a conciliatory manner, was well received by the authorities in Zurich. Bullinger then switches to German and tells Myconius about an incident in Lucerne where a hermit had murdered a number of people and a loud bang could be heard afterwards which was interpreted as a divine warning. Again, the models trained on or including the Early New High German texts do not assign any labels. The regest-only model assigns the tags *Daily Life*, *Evangelical-Reformatory Movement*

[5]HBBW 2492.

[6]HBBW 749.

and *Networking*. *Daily Life* is correctly assigned, as reports about criminal affairs fall under this label, as well as reports about divine signs. *ERM* is also correctly attributed due to the short part about the Helvetic Confession. I would not label this document with *Networking*, as nothing in the text really refers to such a thing besides the expressed gratitude. I believe the labels in this case to be useful, but it does demonstrate what advantage more fine-grained labels could offer. *Daily Life* is a very broad label, and some sub-labels like *Crime*, *Travel*, *Livinghood* would probably be useful for researchers. Of course this could be done a second classification algorithm instead, which would only label those letters which had *Daily Life* assigned to them in the first classification step. The label *Conflict* would also fit this document, but in the training data, this label is mostly used to refer to theological disputes, which is why it is not assigned here.

The third example letter is signed by some members of the Zurich council responsible for the college of Zurich and addressed at two teachers at the university of Tübingen. They report that Zurich has founded two colleges where talented and poor youth shall be educated. Some of these young men, who were the bearers of the letter, were now ordered to study in Tübingen and the council requests the teachers in Tübingen to receive them well, educate them and care for them. The letter continues to detail what these students should learn. The letter closes with the expectation that the students will not be discriminated due to them being Swiss, and assures the teachers from Tübingen that any Wurttembergian students who would come to Zurich would be welcomed in the same manner. Like before, the models incorporating the original german texts do not assign any labels to the document. The regest-only model assigns the labels *Daily Life*, *Education* and *Networking*. The label *Education* is obviously correctly assigned here. *Daily Life* is difficult to explain, if one only looks at the regest, which is all the algorithm sees in this case. But inspecting the original text, there is a whole paragraph about the funding of the studies, which falls under the label. The regest summarises this paragraph shortly with a request for the addressees to manage the finances of the students and care for them. Even though the system could have not known about the whole paragraph, the label is correctly assigned. The few signal words like "Stipendiengelder" (Scholarship funds) and "Lebenswandel" (Lifestyle) might have been enough to assign *Daily Life*. Again, the label *Networking* was assigned, which is correct in this instances, as the recommendation of students as well as making requests to other people is considered to be part of the *Networking* label. The labels for this document are very useful and coupled with the information contained in the metadata and found named entities, a researcher has a good indication what this document is about before even reading the regest.

After examining the assigned labels on these examples, and some others which are

not reported here, I can state that most documents are labelled in a way which is at least helpful to get a broad understanding what a letter is about. Especially the labels *Military Conflicts*, *Realm Politics*, *World Affairs* (not present in these examples), *Education* and *Daily Life* were assigned in a reasonable and useful manner. *Culture of Correspondence* and *Networking* suffer from the fact that they get assigned to most letters. Sadly the models using the Early New High German texts do not produce any labels which is probably due to the minimal preprocessing, which only consists of tokenisation and removal of words below the minimum document frequency threshold. Further research could continue by expanding here and with proper preprocessing and some standardisation of the Early High German Texts, they can probably become a useful resource as well. It is already promising that the regests by themselves produce useful labels. This does of course limit the classification potential to only those letters with regests.

6 Conclusion

In this work I tested a number of different methods to classify documents. These experiments showed that topic modelling as well as TF-IDF are useful methods of feature extraction for Latin and modern German texts.
An interesting observation was the advantage of using the modern German data set over the original Latin texts. Firstly, the success of the extended Latin data set compared to the only-Latin data set demonstrated that even if only some of the documents contain regests, it would help the overall performance. The fact that the modern German data set performed even better shows that due to the bag-of-word approach of the feature extraction methods, even partially incorrect translations of the texts are better than the original texts, if it helps to reduce the data set to a single language instead of having multiple languages that need to be combined in a topic model. Especially the topic modelling algorithms profited from this unification, while TF-IDF performed well on the mixed-language data set as well. In the experiments reported in this paper, I had translations for only a part of the complete collection. Further research could be conducted to investigate if more data would further improve the feature extraction, or if the currently available data has already reached a point where more data does not improve the system. Using more data might even show worse results, because an advantage of the current smaller data set which contains translations is that it contains documents from a similar time frame, from the first 20 years of the collection. Newer letters could introduce new topics which might make the feature extraction worse. Of course, if the system should be applied to the whole collection, using all data when training the topic models or generating the TF-IDF vectors is necessary in any case.
An unexpected result was that TF-IDF performed almost as well as the topic modelling when dimensionality reduction was applied. It is unclear to me why I was not able to reproduce the results which previous research had reported that showed topic modelling being much better at classifying documents than TF-IDF. A possible reason could be the strong mix of topics in the documents of the Bullinger collection which is unusual compared to common evaluation corpora like Wikipedia or collections of news articles, which are more monothematic.
In the preprocessing step, I could observe that some methods of preprocessing worked

better with certain feature extraction methods, such as TF-IDF working better with a lot of filtering of low frequency words (including only words with more than 50 occurrences) while Correlated Topic Models worked better if the filter was only applied to words which occurred less than 25 times. In the stop word filtering it was visible as well that CTM preferred a less strict filtering.
The investigation of the impact of different settings for the creation of the topic models gave some interesting insights, although the randomness inherent to the topic models was an obstacle. Still, some trends could be observed, which were in line with what previous research had shown, such as model performance getting worse if too few or too many topics are defined. Other settings, such as the choice of the alpha prior, did not paint a clear picture which topic models performed better. As mentioned before, it might still be most pragmatic to train a number of models with different settings multiple times to find topic models that perform extraordinarily well.
Classifiers showed better performance with certain feature extractions, with the Multi Layer Perceptron classifier working much better with TF-IDF due to the low dimensionality of the training vectors, while the Support Vector Machine coupled with One-Vs-Rest or Classifier Chain would outperform the inherently multilabelable classifiers in all other cases.
The final results are disappointing to me, but at the same time promising. I had hoped for better results, especially on the labels which were not assigned at all, but the performance on those labels which were assigned a lot, such as *Evangelical-Reformatory Movement*, *Military Conflicts* or *Realm Politics*, demonstrates the potential that these methods offer. An expansion of the training data would be my first step to a better performance. The potential for human interpretability which was discussed in section 5.6.1 should also not be disregarded. A revision of the label set could also be discussed to produce better distinguishable labels, such as combining *World Affairs* and *Realm Politics* into one, and discarding labels which largely depend on single words that can better be found through a keyword search, such as *Islam* and *The Plague*.
This work did not pose the problems which I expected to encounter when I started. I expected that the main obstacles would be due to the fact that I was working with medieval letters, but instead, it was the fact that the classification process was a multi-label process, which is less common in previous research and available implementations. My main regret is that I was not able to employ one of the new topic modelling methods which involve using word embeddings, which would also be a logical next step when continuing research in this direction.

References

Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459.

Blei, D. M. and Lafferty, J. D. (2005). Correlated topic models. *Proceedings of the 18th International Conference on Neural Information Processing Systems (held in Vancouver, British Columbia, Canada)*, page 147–154.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(1):993–1022.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory (held in Pittsburgh, Pennsylvania, USA)*, pages 144–152.

Bächtold, H. U. (2011). Bullinger, Heinrich. In *Historisches Lexikon der Schweiz (HLS)*. [Online; posted 07.04.2011; visited 15.04.2022].

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J., and Blei, D. (2009). Reading tea leaves: How humans interpret topic models. *Proceedings of the 22nd International Conference on Neural Information Processing Systems (held in Vancouver, British Columbia, Canada)*, page 288–296.

Cunningham, P. and Delany, S. J. (2021). K-nearest neighbour classifiers - a tutorial. *Association for Computing Machinery: Computing Surveys*, 54(6):1–25.

Das, R., Zaheer, M., and Dyer, C. (2015). Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (held in Beijing, China)*, volume 1, pages 795–804.

Dieng, A. B., Ruiz, F. J., and Blei, D. M. (2020). Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Dumont, M., Marée, R., Wehenkel, L., and Geurts, P. (2009). Fast multi-class image annotation with random subwindows and multiple output randomized trees. In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications (held in Lisboa, Portugal)*, volume 2, pages 196–203.

Fischer, L., Scheurer, P., Schwitter, R., and Volk, M. (2022). Machine Translation of 16th Century Letters from Latin to German. In *Proceedings of LT4HALA 2022 - 2nd Workshop on Language Technologies for Historical and Ancient Languages (held in Marseille, France)*.

Gao, W., Peng, M., Wang, H., Zhang, Y., Xie, Q., and Tian, G. (2019). Incorporating word embeddings into topic modeling of short text. *Knowledge and Information Systems*, 61(2):1123–1145.

He, J., Hu, Z., Berg-Kirkpatrick, T., Huang, Y., and Xing, E. P. (2017). Efficient correlated topic modeling with topic embedding. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (held in Halifax, Nova Scotia, Canada)*, pages 225–233.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (held in Berkeley, California, USA)*, pages 50–57.

Johnson, K. P., Burns, P. J., Stewart, J., Cook, T., Besnier, C., and Mattingly, W. J. B. (2021). The Classical Language Toolkit: An NLP framework for pre-modern languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations (held online)*, pages 20–29.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. (2019). Text classification algorithms: A survey. *Information*, 10(4):150.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (held in Beijing, China)*, volume 2, pages 1188–1196.

Lenc, L. and Král, P. (2016). Deep neural networks for Czech multi-label document classification. In *Computational Linguistics and Intelligent Text Processing. 17th International Conference in Konya, Turkey, Revised Selected Papers*, volume 2, pages 460–471.

Li, C., Wang, H., Zhang, Z., Sun, A., and Ma, Z. (2016). Topic modeling for short texts with auxiliary word embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (held in Pisa, Italy)*, pages 165–174.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, Scottsdale, Arizona, USA, Workshop Track Proceedings*.

Newman, D., Asuncion, A., Smyth, P., and Welling, M. (2009). Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10:1801–1828.

Piotrowski, M. (2012). Natural language processing for historical texts. *Synthesis lectures on human language technologies*, 5(2).

Qi, P., Zhang, Y., Zhang, Y., Bolton, J., and Manning, C. D. (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 (held online)*, pages 101–108.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (held in Singapore, Republic of Singapore)*, pages 248–256.

Ramage, D., Manning, C. D., and Dumais, S. (2011). Partially labeled topic models for interpretable text mining. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (held in San Diego, California, USA)*, pages 457–465.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359.

Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining (held in Shanghai, China)*, pages 399–408.

Saif, H., Fernandez, M., He, Y., and Alani, H. (2014). On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (held in Reykjavik, Iceland)*, pages 810–817.

Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (held in Berlin, Germany)*, volume 1, pages 86–96.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Tang, J., Meng, Z., Nguyen, X., Mei, Q., and Zhang, M. (2014). Understanding the limiting factors of topic modeling via posterior contraction analysis. In *Proceedings of the 31st International Conference on Machine Learning (held in Beijing, China)*, volume 1, pages 190–198.

Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. (2009). Evaluation methods for topic models. In *Proceedings of the 26th International Conference on Machine Learning (held in Montreal, Quebec, Canada)*, pages 1105–1112.

Wang, W., Huang, Y., Wang, Y., and Wang, L. (2014). Generalized autoencoder: A neural network framework for dimensionality reduction. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (held in Columbus, Ohio, USA)*, pages 496–503.

Xun, G., Li, Y., Zhao, W. X., Gao, J., and Zhang, A. (2017). A correlated topic model using word embeddings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (held in Melbourne, Australia)*, pages 4207–4213.

Zuo, Y., Wu, J., Zhang, H., Lin, H., Wang, F., Xu, K., and Xiong, H. (2016). Topic modeling of short texts: A pseudo-document view. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (held in San Francisco, California, USA)*, pages 2105–2114.

A Tables

Tag	Precision	Recall	F-Score
Conflicts	0.7810	0.2378	0.3483
Culture of Correspondence	0.6899	0.6640	0.6703
Daily Life	0.7018	0.5173	0.5797
Ecclesiastical Organisation	0.0000	0.0000	0.0000
Education	0.9100	0.3063	0.4488
ERM	0.8207	0.7294	0.7712
Humanism	0.4000	0.0508	0.0900
Islam	0.0000	0.0000	0.0000
Military Conflicts	0.8250	0.7419	0.7772
Networking	0.6752	0.9186	0.7758
Personal Affairs	0.0000	0.0000	0.0000
Realm Politics	0.8505	0.6755	0.7449
Religious Persecution	0.6000	0.1464	0.2338
Roman-Catholic Church	0.0000	0.0000	0.0000
Swiss Confederacy	0.2000	0.0154	0.0286
The Plague	0.0000	0.0000	0.0000
World Affairs	0.7648	0.5708	0.6491
Average	0.4835	0.3279	0.3599

Table 10: Individual label scores achieved by the best topic model.

B Data Examples

B.1 Standardised Project Format

```
<letter id="11">
    <metadata>
        <senders>
            <sender id="465"/>
        </senders>
        <addressees>
            <addressee id="495"/>
        <addressees>
        <date>1525</date>
        <language>la</language>
        <translation>True</translation>
        <annotated>True</annotated>
    </metadata>
    <regest>Lorem ipusm</regest>
    <text>
        <p>
            <s>Lorem Ipsum</s>
        </p>
    </text>
    <translated_text>
        <s>Lorem Ipsum</s>
    </translated_text>
    <annotation>
        <tag>Lorem ipsum</tag>
    </annotation>
</letter>
```

Lebenslauf

Persönliches

Nachname:	Prada Ziegler
Vorname:	Ismail
Adresse:	Dorfstrasse 19b, 5210 Windisch
Telefon:	076 417 04 41
E-Mail:	ismail.prada@gmail.com
Geburtsdatum:	30. Dezember 1993
Zivilstand:	verheiratet, 1 Kind

Arbeitserfahrung

06/2019 - HEUTE	Studentischer Mitarbeiter für Computerlinguistik an der ETH-Zürich. Zuständig für automatisierte Textanreicherung der Zeitschriftensammlung e-perioda.
10/2017 - 08/2021	Wissenschaftliche Hilfskraft für Computerlinguistik am Historischen Seminar der Universität Zürich im Editionsprojekt Königsfelden.
09/2016 - 04/2019	Wissenschaftliche Hilfskraft für Informatik am Romanischen Seminar der Universität Zürich.

Erfahrung in Projekten

06/2017 - 12/2017	*Facharbeit Informatik:* Experimente mit Word-Clustering mit Python3 und sklearn als Methode zur automatischen Erkennung von Datumsangaben in spätmittelalterlichen und frühneuzeitlichen Dokumenten.
05/2017 - 12/2017	*Automatische Namenserkennung auf Achitekturzeitschriften des 19. und 20. Jahrhunderts:* Kooperationsprojekt mit der ETH-Bibliothek. Betreut durch Prof. Martin Volk vom Institut für Computerlinguistik der UZH habe ich hierbei OCR-Korrektur der gescannten Texte durchgeführte, einen Goldstandard erstellt und die Namenserkennungsskripte des Text+Berg-Projekts auf Python3 modernisiert und für die Namenserkennung auf den Architekturzeitschriften angepasst. Das Paper dazu ist öffentlich einsehbar unter https://www.o-bib.de/article/view/5382/7420.
09/2016 - 01/2017	*Programmiertechniken der Computerlinguistik 3:* Gruppenprojekt zur Durchsuchung und Visualisierung eines Korpus. Mein Fokus lag bei diesem Projekt auf der Verwaltung und Durchsuchung der Datenbank.

Ausbildung

08/2018 - HEUTE	Masterstudium an der Universität Zürich in den Fächern Computerlinguistik und Geschichte.
09/2013 - 07/2018	Bachelorstudium in Geschichte (Hauptfach), Computerlinguistik (Grosses Nebenfach) und Informatik (Kleines Nebenfach). Abschluss: Bachelor of Arts UZH
11/2012 - 07/2013	Militärdienst bei der ABC Abwehr Kompanie: Abschluss: Wachtmeister (Gruppenführer) im Bereich Labor im Spezialgebiet Nuklear.
08/2008 - 06/2012	Seeland Gymnasium Biel Schwerpunktfach: Chemie/Biologie Ergänzungsfach: Geschichte Abschluss: Schweizerische Maturität

Universität Zürich[UZH]

Philosophische Fakultät
Studiendekanat

Universität Zürich
Philosophische Fakultät
Studiendekanat
Rämistrasse 69
CH-8001 Zürich
www.phil.uzh.ch

Selbstständigkeitserklärung

Hiermit erkläre ich, dass die Qualifikations-Arbeit von mir selbst ohne unerlaubte Beihilfe verfasst worden ist und dass ich die Grundsätze wissenschaftlicher Redlichkeit eingehalten habe (vgl. dazu: http://www.uzh.ch/de/studies/teaching/plagiate.html).

Windisch, 31.05.2022 .....

Ort und Datum Unterschrift