The Creation of an hierarchical Category Set as Gold Standard for Letterhead Elements

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Third Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

1 Introduction and Motivation

2 Used Data

The documents, which form the basis of this work, are old authorities correspondences where it comes to the storage of nuclear waste. The topics are mainly plans of emergency measures, eliminating access waters and the closure of individual chambers of the mine. At the beginning there were 71 authorities correspondences. 33 of these documents are freely available online. The remaining 38 documents have been provided as part of a research project. The freely accessible documents were written in the years 2009 to 2013. The others, as far as recognizable, are from the years 1971 to 1975. Although picture elements are included in the OCR output, they aren't used in this work. Twelve of the documents were removed because they effectively did not contain any information content. Thus were still 59 documents left with a total of 1239 text areas. In addition, seven empty text areas were removed. Thus remained in 1232 text areas for the following steps. The resulting in the OCR process documents were available in HTML format and has been brought in a CSV format for the next steps which contains the following information for each text area:

- Document name and index the text area within the document
- identifier of the text area assigned classification
- position of the text area in the document and its width
- by the OCR process recognized text

3 Creation of the hierarchical Category Set

The hierarchical category system extends the seven basic classes, which were used by a project partner for the first classification. The hierarchical structure also offers the possibility that instances of individual classes can be summarized in its superclass. In the following work, the documents were processed by two raters. After careful review of the documents by both raters, five subclasses were added to the first hierarchical category system. Thus, the classification model, which was used for the next steps emerged. To generate the training data each text area of the 59 documents were individually classified by the raters. In order not mutually influence each other in the award of the label, the classification of the two raters was done independently after a brief discussion about the meaning of each individual classes. In order to verify if and how well the classification matches of both raters, their agreement was calculated by Cohen's Kappa for each evaluation run. In the first evaluation run a κ value of 0.6993 was achieved.

Nevertheless, it became clear that the hierarchical category system was not yet well enough adapted to the available data. Many of the predetermined classes were not at all or only very rarely assigned. This applies to all subcategories of the third party for example. The third party referred to any personal or address information within a document, which can not be assigned to the actual sender or recipient. There were not only deleted subcategories but also added new ones. This was in the category "(main) text"the case. This was assigned in 41.23 % of the text areas. Therefore, this category has been split into the two sub-categories "content" and "structural elements".

The second review cycle revealed that the addition of the two sub-categories "content" and "structural elements" to the category "(main) text"

simplifies the assignment of many text areas. Text areas that were assigned as generic "letterhead element" (i.e. the root category) because they only contained a single character and were used to structure the text, now can be assigned to the category "structural element". This meant that almost 40 % of the previously generically as "letterhead element" marked text areas could be assigned to another category. Overall, the classification of the raters differed after the second pass on the entire data 324 times. This shows that even before the subsequent merging of the data was an agreement of almost 74 % and a κ 0.7007 from among the raters. Compared to the previous run thus a minimum increase of κ is observed. In Bakeman et al. (1997) is shown that a better value for κ is obtained by a larger number of classes. That in this evaluation run now, despite a lower number of classes (37 instead of 47) an increase of κ is observed, therefore, indicates a better match of the raters.

In order to create a unique category assignment for this data on the basis of the final hierarchical category system, the text areas whose assignments were different in the two raters, were checked again together and a category was chosen. Overall, the decisions, which of the two views was followed out very evenly distributed. On the results of the merge you can see that only the upper categories of the category system were not assigned. It follows that is at least one example of the training data available for each of the categories.

4 Classification Experiments

Da innerhalb dieser Trainingsdaten einige Klassen häufig mehrmals hintereinander vergeben wurden und die Implementation die Teilmengen für die k-fold Cross-Validation anhand ihres Index zusammenfügt, wurden die vorliegenden Daten in eine zufällige Reihenfolge gebracht, um eine bessere Verteilung der einzelnen Testfälle in den Trainingsdaten zu gewährleisten. Die ersten Testergebnisse zeigen, dass durch Klasssifikatoren wie den Maximum Entropie Klassifikator oder den Balanced Winnow Klassifikator bereits gute Werte in der Accuracy im Bereich von 0.7 erzielt werden.

Wie bereits in Abschnitt **??** erwähnt, erreichte der C4.5 Klassifikator trotz seiner längeren Laufzeit lediglich einen Accuracy Wert der unter dem Durchschnitt der restlichen Klassifikatoren liegt. Auch die Decision Tree und Winnow Klassifikatoren erzielten unbefriedigende Werte, weshalb im weiteren Verlauf nur auf den Naive Bayes Klassifikator, Maximum Entropie und Maximum Entropie L1 Klassifikator und den Balanced Winnow Klassifikator eingegangen wird.

Um jedoch genauere Aussagen über die Güte eines Klassifikators im Hinblick auf einzelne Klassen zu erhalten, wurden zusätzlich zur Accuracy jedes Klassifikators noch die Precision und Recall Werte jeder einzelnen Klasse durch eine Erweiterung der bestehenden Implementation berechnet. Bei der Cross-Validation erfolgt die Aggregation oft über das Summieren der Precision und Recall Werte für die einzelnen Durchläufe und eine Mittelwertbildung über denselben. Zunächst wurde dieses Verfahren durch die Verwendung der Trail Klasse von $MALLET$ implementiert. Hierbei wurden die getF1(), getPrecision() und getRecall() Methoden verwendet. Da ein Trial im Verfahren der 10-fold Cross-Validation jedoch durch einen Split der Daten in neun Trainings- und einen Testdatensatz erzeugt wird, besteht ein Testdatensatz nur aus 132 oder 133 Textbereichen. Das hierarchisches Kategoriensystem besteht aber bereits aus 37 Klassen, wovon 32 im Rahmen der Trainingsdatenerzeugung vergeben wurden. Da die Klassen innerhalb des Trainingskorpus nicht gleichmäßig verteilt sind, ist ein Vorkommen jeder Klasse in jedem Datensatz sehr unwahrscheinlich. Darüber hinaus existieren Klassen, die sogar weniger als 10 mal vergeben wurden und folglich nicht in jedem Datensatz vorkommen können.

Stattdesssen wurde in der vorliegenden Arbeit die Aggregation über das Summieren der Confusionmatrizen der einzelnen Validierungsläufe, welche durch die Verwendung der ConfusionMatrix Klasse von $MALLET$ erzeugt wurden, zu einer Gesamtmatrix vorgenommen. Die Werte für F1, Precision und Recall wurden auf Basis dieser Gesamtmatrix berechnet.

Eine stichprobenartige Überprüfung zeigte, dass zahlenmäßig häufig auftretende Klassen wie zum Beispiel „Fließtext" (1421) oder „Sender(Adresse, Postfach" (1123) eine geringe Abweichung der F1 Werte von wenigen Prozentpunkten (im Bereich von circa plus zwei bis minus sechs Prozent) von den mittels getPrecision() und getRecall() berechneten F1 Werten aufweisen. Dabei besitzen die durch die Aggregation der Confusionmatrizen berechneten F1 Werte, in der Regel, den geringeren Wert. Das hat den positiven Nebeneffekt, dass die Leistungsfähigkeit unserer

Klassifikatoren eher unterschätzt als überschätzt wird. Dieser positive Aspekt ist auch bei zahlenmäßig selten auftretenden Klassen wie zum Beispiel „Datum“ (121) oder „Ort“ (13) zu beobachten.

To check whether the manually assigned class to the training data can be reproduced by machine, we trained several training algorithms to check, which one would get the best results for the classifikation of future documents. The classifiers we used are the Naive Bayes, Maximum Entropy, Maximum Entropy L1, and Balanced Winnow classifiers. We used Winnow, C4.5 and Decision Trees at the beginning but omitted those in the further steps because they eiter achieved a way lower accuracy or needed a far longer training time (about 27x) without better results. For the implementation of the training algorithms we used the java based $MALLET$ toolkit.

Because within the training data, some classes were often repeatedly awarded and the implementation joins the subsets for k-fold cross-validation based on their index, the order within the available data were randomized to create a better distribution of the individual test cases in the training data.

The first test results show that good results in accuracy in the range of 0.7 already be achieved by the maximum entropy classifier as Klasssifikatoren or the Balanced Winnow classifier.

Bei der Verwendung des hierarchischen Kategoriensystems zeigte sich bereits, dass trotz der relativ guten Accuracy manche der Klassen sehr schlechte Werte bezüglich Precision und Recall aufweisen. Dies tritt sehr häufig bei Klassen mit geringem Vorkommen auf. Auch wird deutlich, dass der Naive Bayes Klassifikator die meisten Probleme mit der geringen Anzahl von Trainingsbeispielen hat. Grundsätzlich wird aber auch klar, dass Klassen die insgesamt weniger als 25 mal im Trainingsdatensatz vorkommen häufig schlecht erkannt werden. Diese Grenze fiel bereits bei der Verwendung der getPrecision() und getRecall() Methoden von $MALLET$ auf (s.o.). Eine Ausnahme hierzu bilden die Klassen „Generische Begrüßung“ und „Begrüßung, Name“. Obwohl sie nur 11 bzw. 8 mal vorkommen, besitzen sie bei allen vier Klassifikatoren hohe Werte für Precision und Recall die im Bereich von 0.9 bis 1.0 liegen. Auch haben manche Klassen zwar eine hohe Precision aber einen sehr geringen Recall.

Beim Vergleich der einzelnen Klassifikatoren wird deutlich, dass der Naive Bayes Klassifikator die meisten Probleme mit den vorliegenden Daten hat. Sein Accuracy Wert liegt 0.05 bis 0.07 unter dem der Anderen. Während die restlichen drei Klassifikatoren für sechs oder sieben Klassen einen F1 Wert von null erreichten, war dies beim Naive Bayes Klassifikator bei 13 Klassen der Fall. Das heißt für 13 zugegebenermaßen geringfügig vorkommenden Klassen der insgesamt 32 vergebenen Klassen leistet der von uns trainierte Naive Bayes Klassifikator effektiv nichts. Auch bei den restlichen Klassen erzielte der Naive Bayes Klassifikator meistens schlechtere Ergebnisse. Ausnahmen hierbei bilden lediglich die Klassen „Briefkopfelement“ und „Sender (Telefon/Fax)“. Für die Klasse „Sender (Email/Webseite)“ erreichte nur der Maximum Entropie L1 Klassifikator einen besseren Wert als der Naive Bayes Klassifikator.

Beim weiteren Vergleich der Klassifikatoren fällt auf, dass der Balanced Winnow Klassifikator als einziger einen Wert größer als null für die Klasse „Datum etc.“ erzielt. Trotz eines geringen Vorkommens von nur fünf Textbereichen in den Trainingsdaten liegt der F1 Wert des Balanced Winnow Klassifikators für diese Klasse bei 0.889. Alle Ergebnisse der einzelnen Klassifikatoren dieses Durchganges sind in den Tabellen **??** bis **??** in den Anlagen aufgelistet.

Wie bereits in Abschnitt **??** beschrieben, besteht ein Vorteil eines hierarchischen Kategoriensystems darin, dass jede Instanz jeder Klasse auch zu jeder Oberklasse der betreffenden Klasse gehört. Dadurch können Klassen mit einem geringen Vorkommen zu ihrer Oberklasse hinzugefügt werden, ohne die intensionale Definition der Klasse wesentlich zu verändern. Diese Eigenschaft ermöglicht es, durch die Abwandlung des bestehenden hierarchischen Kategoriensystems Trainingsdatensätze mit abgewandelten Klassenlabeln zu erzeugen. In unserem Fall wurden drei weitere Versionen der Trainingsdatensätze durch das Kollabieren des Kategoriensystems auf drei Ebenen,

zwei Ebenen und eine ebenenunabhänige Version erzeugt, bei der jede Klasse, deren Vorkommen im Trainingsdatensatz geringer als 19 war, in ihre Oberklasse gemappt wurde. Die Grenze von 19 stellt dabei den Median der Klassenstärken dar, also den Median der Anzahlen der den verschiedenen Klassen zugeordneten Textbereichen in den Trainingsdaten (diese letzte Version wird im weiteren Verlauf dieser Arbeit der Einfachheit halber als Median Version bezeichnet). Durch dieses Verfahren wird überprüft, inwiefern die Werte für Precision und Recall der einzelnen Klassen sich durch die zusätzlichen Elemente verändern. Grundsätzlich wird davon ausgegangen, dass sich der Accuracy Wert durch die geringere Anzahl von Klassen verbessert, da bei der Klassifikation zwischen weniger Klassen unterschieden werden muss. Das Verhalten von Precision und Recall ist hierbei schwer einzuschätzen, da bei einer größeren Anzahl von Trainingsbeispielen einerseits davon auszugehen ist, dass sich Precision und Recall verbessern. Jedoch können die zusätzlichen Elemente auch dazu führen, dass die charakteristischen Merkmale der Klasse nicht mehr so stark ausgeprägt sind, da die zusätzlichen Elemente von den Ursprünglichen abweichen. In Abbildung **??** wird gezeigt, dass sich für das normale Mapping pro Ebene die Accuracy Werte für jeden Klassifikator verbessern. Überraschenderweise fällt bei der Betrachtung der Accuracy Werte auf, dass bei der Verwendung des Median gemappten hierarchischen Kategoriensystems, welches deutlich weniger Klassen enthält als das normale hierarchische Kategoriensystem, kaum eine Steigerung erreicht wird. Der Accuracy Wert für Balanced Winnow sinkt sogar im Vergleich zum Ausgangswert.

Zusätzlichen zu den verschiedenen Versionen des hierarchischen Kategoriensystems wurde ein Verfahren verwendet, welches bereits in Preßler (**?**) zur Anwendung kam. Hierbei werden für jeden Textbereich zusätzliche Features (im folgenden X-Features genannt) seinem Inhalt entsprechend hinzugefügt. Der Aufbau dieser zusätzlichen Features ist abhängig von den bereits enthaltenen Features eines Textbereiches. Für jedes Element des Textbereiches (d.h. im wesentlichen für jedes enthaltene Wort) wird ein zusätzliches Element hinzugefügt. Somit verdoppelt sich die Anzahl der Features pro Textbereich. Die Form der zusätzlichen Features ergibt sich aus daraus, dass zwischen Wörtern bzw. Zeichen und Zahlen differenziert wird. Wörter werden zusätzlich durch ihre Groß- und Kleinbuchstaben unterschieden. Ein großes X im zusätzlichen Element steht hierbei für einen Großbuchstaben. Analog hierzu steht ein kleines x für einen kleinen Buchstaben. Somit ergibt sich für das Wort „Bundesamt“ die Zeichenkette „Xxxxxxxxx“. Zahlen werden hierbei etwas anders behandelt. Sie beginnen mit einem großen X gefolgt von einer Anzahl von großen N, welcher der Länge der Zahl entspricht. Somit ergibt sich für die Zahl „53175“ die Zeichenkette „XNNNNN“. Diese Konvention wurde bereits in Preßler (**?**) verwendet. Das Ziel dieser zusätzlichen Features besteht darin, Wörter oder Zahlen die ein ähnliches Format besitzen (beispielsweise Postleitzahlen) stärker miteinander in Verbindung zu bringen als dies durch die nominelle Inklusion der ursprünglichen Zeichenkette als Features in einem Bag-of-Words Modell geschehen kann. Möglicherweise können die Klassifikationsergebnisse dadurch verbessert werden. In der vorangehenden Arbeit zeigte sich allerdings, dass die Verwendung dieser X-Features nicht den gewünschten positiven Effekt erzielte. Die meisten Klassifikatoren reagierten in Preßler (**?**) eher mit einer schlechteren Accuracy als zuvor. Eine geringe Verbesserungen der Accuracy konnte lediglich bei den Maximum Entropie L1, Decision Tree und C4.5 Klassifikatoren gemessen werden. Um zu überprüfen, ob durch die Verwendung der X-Features auf unserem Datensatz eine Verbesserung der Ergebnisse der Klassifikatoren erreicht werden kann, wurden die Trainingsalgorithmen auf einer Kopie des zuvor verwendeten Datensatzes, welcher die zusätzlichen X-Features enthält, trainiert. Dabei zeigte sich, dass die Accuracy des Naive Bayes Klassifikator um bis zu 0.07 sinkt. Dies überrascht nicht, da dieser Klassifikator, wie bereits in Abschnitt **??** beschrieben, schlecht mit ähnlichen Features umgehen kann. Die Auswirkung auf die anderen Klassifikatoren ist nicht so schwerwiegend. Beim Maximum Entropie Klassifikator wurde durch die Verwendung der X-Features in drei von vier Fällen eine Verbesserung der Accuracy von bis zu 0.014 festgestellt. Lediglich unter Verwendung des Kategoriensystems mit 2 Ebenen wurde ein minimal geringerer Wert erzielt (0.856 ohne X-Features und 0.852 mit X-Features). Für den Maximum Entropie L1 Klassifikator fallen die Ergebnisse hierbei ähnlich aus. In zwei von vier Fällen ver-

besserte sich die Accuracy, während sich bei den anderen entweder kein oder nur ein sehr geringer Verlust bemerkbar machte. Ein anderes Ergebnis kann man beim Balanced Winnow Klassifikator beobachten. Dieser erreichte ohne die X-Features in drei von vier Fällen ein minimal besseres Ergebnis als mit ihnen. Grundsätzlich lässt sich sagen, dass in den meisten Fällen die Verwendung der X-Features kaum merkliche Unterschiede erzeugt. Die Unterschiede in den Accuracy Werten sind in Tabelle **??** zusammengefasst.

Bei der Betrachtung des Verhaltens der Klassifikatoren unter Verwendung der reduzierten Kategoriensysteme fällt beim Naive Bayes Klassifikator auf, dass die Ergebnisse für Sender und Empfänger sehr unterschiedlich ausfallen. Die Klassen des Senders werden einzeln meistens besser klassifiziert als wenn man sie in eine Klasse zusammenfügt. Beim Empfänger ist das umgekehrt. Für diese Klasse wurden die besten Ergebnisse erzielt, wenn man sie zu einer Klasse zusammenfügt. Diese Beobachtung kann man auch bei den anderen Klassifikatoren machen. Die einzige Ausnahme hierzu stellt der Maximum Entropie L1 Klassifikator dar. Mit ihm werden für alle Empfängerklassen ähnlich gute und teils auch bessere Ergebnisse für den F1 Wert erzielt (außer für „Empfänger(Telefon/Fax)“, was aber durch das totale Vorkommen der Klasse von eins im Datensatz irrelevant ist).

Auch fällt auf, dass 19 der 32 vergebenen Klassen im hierarchischen Kategoriensystem mit vier Ebenen weniger als 20 mal in unserem Datensatz vergeben wurden. Um nachvollziehen zu können, wie sehr sich die Anzahl einer Klasse innerhalb der Trainingsdaten auf den F1 Wert auswirkt, wurde in dieser Arbeit für jeden Klassifikator überprüft, wie oft ein Klassifikator einen niedrigen F1 Wert für selten vorkommende Klassen erzielt. Im Vergleich dazu wurde auch überprüft, wie oft dieser Klassifikator einen niedrigen F1 Wert für häufig vorkommende Klassen erzielt. Für das Vorkommen wird eine Grenze von 20 verwendet, da sich die Grenze von 20 bis 25 bereits in früheren Arbeitsschritten dieser Arbeit bemerkbar machte. Es wird deutlich, dass für Klassen, die häufiger als 20 mal im Datensatz vorkommen, deutlich häufiger ein F1 Wert über 0.5 erzielt wird als für Klassen die seltener als 20 mal vorkommen. Auch sieht man, dass der Maximum Entropie L1 und der Balanced Winnow Klassifikator deutlich häufiger F1 Werte über 0.5 bei einer geringen Anzahl von Trainingsbeispielen erzielen als die anderen zwei Klassifikatoren. Die Ergebnisse sind in Tabelle **??** und Tabelle **??** bis **??** dargestellt. Für eine detailliertere Darstellung aller Ergebnisse jedes einzelnen Klassifikators wurden die genauen Ergebnisse für F1, Precision und Recall für jede Klasse in Tabelle **??** bis **??** aufgelistet.

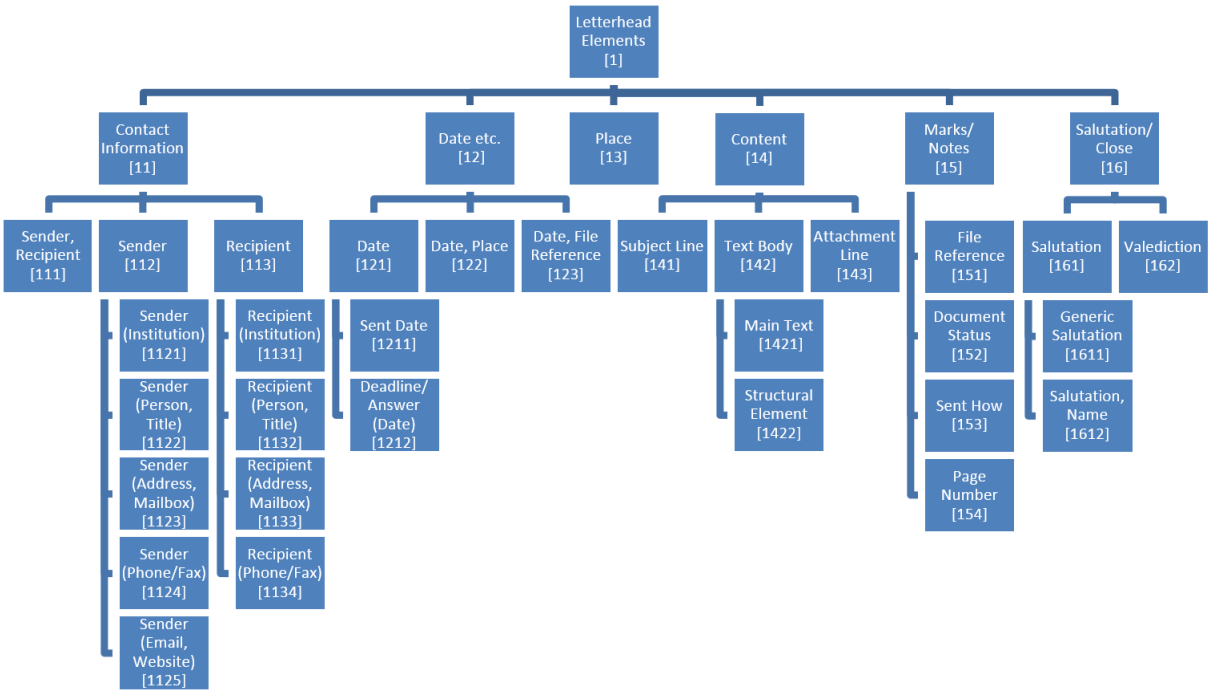
5 Future Work

Acknowledgments

References

Bakeman, R., Quera, V., McArthur, D. und Robinson, B. F.: *Detecting Sequential Patterns and Determining Their Reliability With Fallible Observers*, in: *Psychological Methods*, Vol. 2, S. 357–370, 1997.

Autoren, erster Vorname ausgeschrieben restlichen mit . Personen mit , getrennt. Jahr. *Titel*. Verlag, Verlagsort, Verlagsland. Volume(Item/Number):Seitevon-bis.


Letterhead Elements [1]
Contact Information [11]
Date etc. [12]
Place [13]
Content [14]
Marks/ Notes [15]
Salutation/ Close [16]
Sender, Recipient [111]
Sender [112]
Recipient [113]
Date [121]
Date, Place [122]
Date, File Reference [123]
Subject Line [141]
Text Body [142]
Attachment Line [143]
File Reference [151]
Salutation [161]
Valediction [162]
Sender (Institution) [1121]
Sender (Person, Title) [1122]
Sender (Address, Mailbox) [1123]
Sender (Phone/Fax) [1124]
Sender (Email, Website) [1125]
Recipient (Institution) [1131]
Recipient (Person, Title) [1132]
Recipient (Address, Mailbox) [1133]
Recipient (Phone/Fax) [1134]
Sent Date [1211]
Deadline/ Answer (Date) [1212]
Main Text [1421]
Structural Element [1422]
Document Status [152]
Sent How [153]
Page Number [154]
Generic Salutation [1611]
Salutation, Name [1612]