Advanced Training Set Construction for Retrieval in Historic Documents

Andrea Ernst-Gerlach and Norbert Fuhr

University of Duisburg-Essen, Department of Computational and Cognitive Sciences,
Lotharstr. 65, 47048 Duisburg, Germany
ernst@is.inf.uni-due.de
norbert.fuhr@uni-due.de

Abstract. Retrieval in historic documents with non-standard spelling requires a mapping from search terms onto the historic terms in the document. For describing this mapping, we have developed a rule-based approach. The bottleneck of this method has been the training set construction for the algorithm where an expert has to assign manually current word forms to historic spelling variants. As a better solution, we apply a spell checker on a corpus of historic texts, which gives us a list of candidate terms and associated suggestions. The new method generates possible rules for the suggestions and accepts the most frequent rules. Experimental results with German and English texts from different centuries demonstrate the feasibility of our approach. Thus a training set can be constructed with much less initial effort.

Key words: Spelling variation, training set construction, historic documents

1 Introduction

The number of digital historical collections is continually growing. But even though full text search is available, many documents can not be found because they use a non-standard spelling. E. g. the German word *akzeptieren* is the contemporary word of the spelling variant *acceptieren*. The non-standard spelling produces problems when searching in historic parts of digital libraries. Most users will enter search terms in their contemporary language which differs from the historic language used in the documents.

However, even popular digitization initiatives like Google Book Search[1] or the European Digital Library[2] have not integrated a search for spelling variants yet. In order to solve this problem, our project deals with the research and development of a search engine where the user can formulate queries in contemporary language for searching in documents with an old spelling that is possibly unknown to the user (see [5]).

Other approaches use dictionaries for this purpose (e. g. [7]). However, these approaches cover only the words contained in the dictionary. Furthermore, the time and effort for the manual construction of the word entries is rather high. We overcome this

[1] http://books.google.com/ access: 13.08.2010
[2] http://www.europeana.eu/portal/ access: 13.08.2010

disadvantage with a rule-based approach, in order to be able to cover the complete vocabulary (and thus increase recall). For this purpose, we are developing transformation rules for generating historic spellings from a given word.

Due to the dependency of rules on time and region , rule sets have to be generated over a longer period when suitable corpora become available. In order to get a large rule covering at least 1000 training instances are needed. This work manually has to be done by linguists or historians without help from computer scientists. Thus it is necessary to develop a tool for an easy and fast rule development that does not require computer science knowledge.

In the following, we assume that the user has a new collection and wants to enable a full text search for the documents. Let us further assume that there is no rule set available for the time and region of the collection. One first has to collect evidences consisting of contemporary inflected and derived forms of the lemma (in the following denoted by word forms) and their corresponding historic spelling variants. In a second step, the rules can be developed.

Users may have different interests. For example, for a linguist the creation of evidences is already an interesting research task and thus he wants to create each evidence only with semi-automatic support from the tool since he is interested in the development of the language and wants very precise rules. Possibly, he often searches for all occurrences of a word form in his collection, and thus he can only work with a more complete rule set. By contrast, a historian might only be interested in getting relevant documents. Thus he wants to enable a fuzzy full text search as soon as possible. He might prefer an automatic approach even if he misses some documents in the first step, when he has the chance to improve the rule set later on. Depending on his needs the user will concentrate more on the recall or on the precision of the search. The tool should offer the necessary flexibility at this point. Therefore the user will be offered full support but it will be his choice how many of the suggested evidences (and rules) he is accepting.

The remainder of this paper has the following structure: First, we give a brief survey over related work, and then Section 3 briefly introduces the rule generation process. Section 4 describes how the rule generation algorithm can be used to build evidences and rules automatically. Our approach is evaluated in Section 5, and the last section concludes the paper and gives an outlook on future work.

2 Related Work

Gotscharek et. al. [6] developed LeXtractor, a tool for the construction of historical lexica. The lexicon entries can also be regarded as evidences in our approach. On the one hand the user could work on the lexicon construction based on highlighted unknown terms. On the other hand he can work with a ranked list of unknown terms. In order to rapidly increase the percentage of the tokens from the documents that is covered by the lexicon the list is ordered by decreasing frequency. Because they are used for lexicon construction, the results have to be very precise. Thus the expert has to go through the whole collection and look at each reading of an unknown spelling. As support, a list with so called attestations for an unknown word is offered when it is chosen for the

construction of a lexicon entry. LeXtractor applies manually collected rules (so called patterns) to find the potential contemporary forms in a modern dictionary.

Pilz and Luther [10] developed a method for supporting evidence collection within their Evidencer tool. The Evidencer uses a Bayesian classifier, assuming that the distribution of the n-grams differs significantly between the standard spellings and the non standard spellings. For the separation of unknown words into spelling variants and correct spellings, the classifier estimates the probability of a word being a spelling variant. After the training phase, a list of unknown words is presented which is ranked by decreasing probability of being a spelling variant. The user can adjust the Bayesian classifier by modifying the corresponding probability threshold for possible spelling variants.

VARD 2 developed by Baron and Rayson [2] also finds contemporary word forms for spelling variants in historic documents. The tool marks all words as potential variants that have not been found in a modern lexicon. For each marked word, a ranked list of candidate modern forms is offered to the user. He can then chose the correct modern form for the possible spelling variant. Additionally, a second mode is offered where the tool can automatically accept suggestions. In this mode, for each potential variant the suggestion with the highest ranking is accepted, if the corresponding score is higher than a user-defined threshold value. For providing the suggestions, Baron and Rayson use a manually created evidence list, a modified version of the SoundEx algorithm and manually created replacement rules. Based on these methods the confidence score for a suggestion is generated. This score is not a fixed value. It is adapted after each process step.

The first approach needs a lot of manual interaction for creating the evidences as well as for the rule development, even with the offered support. The second approach looks more promising regarding the automatic support for the user and the possibility for the user to influence the results by a threshold value, but the Bayesian classifier needs a lot of training data as input. Thus a huge amount of manual work is necessary before the classifier can be used. Additionally, the user can only work document-wise; he can not look at several occurrences of unknown terms in different documents at once. The permanently adapted confidence value for the modern word forms from the third approach is remarkable. The confidence score is comparable to the Bayesian classifier of the Evidencer tool. The disadvantage of VARD 2 are the methods which need a training set and a rule set as input. Both sets are manually created. Additionally, the SoundEx algorithm is a phonetic algorithm which has been developed for contemporary English. Thus the approach is not language-independent.

In summary, none of the presented approaches overcomes the bottleneck. All of them need a lot of manual effort, at least in the beginning, in order to initialise the tools. Thus an approach that can automatically detect evidences for a training set will make the access to historic documents much more comfortable for the user.

3 Generation of Transformation Rules

Now we give a brief overview on the methods for evidence collection and rule-generation methods used in the past (see [4]). In order to generate rules for transforming contem-

porary query terms onto the historic spelling variants, we first need a training set. By using a spell checker, we are getting a list of candidate words for historic documents in non-standard spelling. We are using Hunspell as spell checker[3], which currently offers dictionaries for 98 different languages. The suggestions for the misspelled word are generated based on n-gram similarity, rules and pronunciation data based on a dictionary. We have to check manually that the words are actually of a non-standard spelling, and have to assign the equivalent words in the contemporary standard spelling. Furthermore, we determine the number of occurrences of each historic word form. Afterwards, we can focus on the second step — the building of new rules.

The automatic rule generation method starts with a training sample of historic texts. Thus, we have sets of triplets containing the contemporary word forms, their historic spelling variant and the collection frequency of the spelling variant.

First, we compare the two words and determine so-called 'rule cores', the necessary transformations, and also identify the corresponding contexts. For example, for the contemporary word form *enclosed* and the historic word form *inclos'd*, we would get the following 2-element set of rule cores: ((e→i)nclos), (nclos(e→')d).

As a second step, we generate rule candidates for each rule core that also takes account of the context information (e. g. consonant (C) or word-ending ($)) of the contemporary word. If we use the example shown above, we find that among others the following candidate rules are generated: e→ ', ed→ 'd, se→ s', sed→ s'd, Ce→ C', ed$→ 'd$.

Finally, in the third step, we select the useful rules by pruning the candidate set (where we are taking the collection frequency into account) with a modified version of the PRISM algorithm (see [3]).

4 Automatically Accepting Evidences

The last section showed that up to now, the approach required a substantial manual effort at the beginning. Therefore, a major goal is to reduce the initial work by developing an algorithm for building evidences automatically.

The basis for the rule-based approach is the assumption that the spelling variants have a certain amount of regularity. We take this assumption also as basis for automatically accepting evidences. The correct contemporary form is often among the suggestions from the spell-checker. We assume that these regularities between spelling variants and the contemporary forms are much less frequent between variants and false suggestions. Thus our algorithm concentrates on the problem of finding the correct suggestion for a possible variant. We choose the correct suggestions by taking those with more frequent rule candidates.

An evidence is created from an unknown spelling and each corresponding suggestion (see Table 1). We use these evidences as training set, and generate the possible rule candidates. Since we do not want to apply the rules in this step, we are not interested in the different rule candidates and thus consider the rule cores. In this way we get a more distinct distribution of the rules.

[3] http://hunspell.sourceforge.net/ access: 13.08.2010

Table 1. Example training set and generated rule candidates

suggestion	**unknown word**	**rule candidates**
Geschicklichke**it**	Geschicklichke**yt**	$i \rightarrow y$
Ungeschicklichke**it**	Geschicklichke**yt**	$un \rightarrow \emptyset$, $i \rightarrow y$
Gesch**wis**t**er**lichke**it**	Geschi**ck**lichke**yt**	$w \rightarrow \emptyset$, $ster \rightarrow ck$, $i \rightarrow y$
jederze**it**	jederze**yt**	$i \rightarrow y$
jeder**mann**	jeder**zeyt**	$mann \rightarrow zeyt$
derze**itig**	**je**derze**yt**	$\emptyset \rightarrow je$, $i \rightarrow y$, $ig \rightarrow \emptyset$

Table 2. Example for accepting evidences for the rule core $i \rightarrow y$

suggestion	**unknown word**	**procedure**
Geschicklichke**it**	Geschicklichke**yt**	accept
jederze**it**	jederze**yt**	accept
obgle**i**ch	obgle**y**ch	accept
Sonderhe**it**	**In**sonderhe**yt**	mark $i \rightarrow y$ accepted

We are assuming that the more often a rule core appears in different evidences, the higher the probability that it is useful. Accordingly, the precision for evidences based on more frequent rule candidates will also increase. Thus, in each run, the most frequent of the unprocessed rule candidates is accepted. If several rule candidates have the same frequency, substitution rules are preferred, since these rules usually have a higher precision than insertion or deletion ones. E. g., we prefer $i \rightarrow y$ over $s \rightarrow \emptyset$, $\emptyset \rightarrow h$.

After we have accepted a rule candidate, we look at the corresponding evidences (and thus at the suggestions of the spell checker). If the evidence is based only on the accepted rule candidate, the evidence is directly accepted. If it is based on more than one rule candidate, it is accepted, as long as the other rule candidates also have been accepted. Otherwise it is only marked that the rule candidate has been accepted (see Table 2).

Since we have now accepted a suggestion, we are looking at the other suggestions for the spelling variants of the accepted evidence in the next step. We are assuming that a spelling variant has exactly one corresponding modern spelling. Thus we can delete the other evidences. This is a simplification, since Pilz [9] observed already that e. g. the spelling variant *Hunngern* has the two modern spellings *Ungarn* (Hungary) and *Hungern* (starvation). This simplification is needed in order to enable the process of accepting evidences automatically and thus to reduce the manual effort. However, we use it only for the automatically accepted evidences. If an evidence is missed during the automatic process, we assume that the necessary rule for it is created by another evidence. During the deletion process the evidences are also deleted from their other corresponding rule cores. Afterwards the whole process starts again with the most frequent unprocessed rule candidate. For the remaining unknown words, the user has to manually add different contemporary forms for one spelling variant.

The user can influence this process by setting a minimum word length, a minimal number of rule occurrences and a maximal number of rule applications per word. From these choices the historian might prefer a shorter word length, a smaller number of rule occurrences and a higher maximal application of rule cores, in order to achieve a high recall. Thus he can immediately start his search if he wants to. In contrast to that, a linguist might prefer the opposite settings for improving the precision.

The results of the chosen evidences are offered within the user interface in form of a list where the user can confirm the collected evidences. The overview of the evidence pairs will also offer access to attestations of the spelling variants. Thus the user can also look at to corresponding context.

5 Evaluation

As test collection, we used documents from the Nietzsche reception[4] and other smaller collections. The collection contains around 100 documents. For the evaluation we randomly choose 10 documents for each century from the 16th to 19th century in order to consider the time dependency of the approach. Since we assumed that the number of helpful suggestions from the spell-checker is decreasing, we made different runs for each century. In order to demonstrate the language-independence of our approach, we also applied it to 10 randomly chosen documents from the Shakespeare collection.

For each subcollection, we first applied our approach and then we randomly tooked 200 unknown word types for evaluation. For each subcollection we perform different runs with at most one, two or three rule applications and then we calculated recall and precision values based on the number of the minimum rule occurrences (2, 5 and 10). For all runs we set the minimum word length of the unknown terms to five. The results can be found in Tables 3, 4 and 5. We calculated two different precision values. The first variant is based on the overall number of accepted evidences while the second one only considers those evidences that are really spelling variants. As a baseline for this evaluation, we took the first suggestion from the spell-checker for each unknown word and calculated the corresponding recall and precision values.

5.1 Temporal Evaluation

The spell checker offers on average 5.4 suggestions per unknown word for the 16th century, 4.9 for the 17th and 18th century and 4.6 for the 19th century texts. Thus the number of suggestions is slightly decreasing for the more modern words.

Regarding the precision values for the different centuries, we discover that the precision is increasing over the time, with the exception of the 19th century. The percentage of unknown terms is decreasing over time from 0.33 (16th century) to 0.06 (19th century). Additionally, the number of different types is higher in the 18th century in comparison to the 19th century. Thus there are only half as many unknown types for the 19th century than for the 18th century. The resulting smaller training set leads to

[4] `http://www2.inf.uni-due.de/Studienprojekte/Nietzsche/pp2001/die_cd/die_cd.htm`

Table 3. Precision based on all unknown terms

Thresholds		German				English
Rule applications	**Rule Occurrences**	**1500-1599**	**1600-1699**	**1700-1799**	**1800-1899**	**1590-1616**
1	2	0.50	0.56	0.62	0.60	0.42
	5	0.51	0.59	0.72	0.63	0.42
	10	0.52	0.61	0.74	0.65	0.46
2	2	0.48	0.51	0.65	0.55	0.38
	5	0.48	0.57	0.71	0.60	0.40
	10	0.52	0.60	0.74	0.64	0.45
3	2	0.48	0.48	0.53	0.54	0.38
	5	0.50	0.53	0.68	0.58	0.39
	10	0.53	0.57	0.71	0.62	0.41
Baseline		0.35	0.32	0.42	0.40	0.44

Table 4. Precision values restricted to spelling variants

Thresholds		German				English
Rule applications	**Rule Occurrences**	**1500-1599**	**1600-1699**	**1700-1799**	**1800-1899**	**1590-1616**
1	2	0.58	0.63	0.70	0.71	0.48
	5	0.59	0.66	0.82	0.75	0.45
	10	0.57	0.66	0.82	0.76	0.50
2	2	0.60	0.61	0.75	0.65	0.46
	5	0.58	0.66	0.78	0.71	0.46
	10	0.62	0.68	0.79	0.74	0.50
3	2	0.58	0.57	0.62	0.64	0.46
	5	0.61	0.61	0.77	0.68	0.45
	10	0.61	0.64	0.77	0.71	0.46
Baseline		0.39	0.36	0.48	0.43	0.58

Table 5. Recall values for the different parameters

Thresholds		German				English
Rule applications	**Rule Occurrences**	**1500-1599**	**1600-1699**	**1700-1799**	**1800-1899**	**1590-1616**
1	2	0.56	0.62	0.78	0.79	0.74
	5	0.52	0.62	0.74	0.77	0.74
	10	0.46	0.61	0.74	0.75	0.74
2	2	0.65	0.66	0.80	0.81	0.77
	5	0.58	0.65	0.77	0.79	0.77
	10	0.53	0.65	0.77	0.79	0.77
3	2	0.71	0.66	0.86	0.81	0.77
	5	0.63	0.65	0.84	0.79	0.77
	10	0.58	0.65	0.84	0.79	0.77
Baseline		0.64	0.56	0.67	0.70	0.66

the decreasing precision. The recall is also increasing over time, with the exception of three rule applications in the 19th century, and three rule applications with two rules in the 16th century. Therefore, the evaluation shows that the quality of our approach is increasing over time, even though exceptions may occur.

5.2 Restricted Number of Rule Applications

Precision is decreasing in three of four cases for increasing numbers of rule applications. All recall values are increasing. In spite of the small number of runs it becomes obvious that a restriction on the number of rule applications per word is a useful parameter for controlling the quality of automatic evidence collection. At a closer look the increase in precision is only very small if the number of rule application is more restricted.

5.3 Minimal Number of Rule Occurrences

The minimal number of rule occurrences achieves even better results than the restricted number of rule applications. For the German runs only two cases (both for the 16th century) can be found where the precision is decreasing in between.

The improvements are also higher for more recent texts. A look at the evaluation data showed that the differences for the various number of rule occurrences are only limited. In this case the threshold should be higher in order to get a remarkable effect for the precision values. As expected, the recall is decreasing in all cases for the German runs. Noticeable is the recall for English. There is only a difference when just one rule is applied. The minimal number of rules occurrences has no influence on recall. Since approximately three out of four words are found already it seem that the parameter settings for English are well chosen regarding the recall.

5.4 Different Precision Values

The precision for all unknown terms gives us an indication of the number of generated incorrect evidences. Since we are not missing any rules when we generate rules for evidences that are not spelling variants, it is also interesting to look at the precision that is restricted to the spelling variants.

As expected, the precision values are much higher in this case. The highest precision is 0.82 for the 18th century when only one rule is applied and the rule occurs in at least 5 evidences. None of the restricted precision values is lower than 0.57 for German. Thus more than half of the chosen suggestions are contemporary forms. Even if we regard the lowest precision for all unknown terms (0.48), it turns out that nearly every second evidence is correct.

5.5 Baseline

Regarding the German examples, the precision is always clearly better than the baseline. With an exception for the 16th century, the recall is also better than the baseline. For the 16th century the recall can also be outperformed by those runs with less restrictive parameters.

The results for the runs based on the Shakespeare documents show that the recall is always better than the baseline. Regarding the precision based on all terms, we get mixed results. Looking at the precision restricted to the spelling variants, we detected that the precision for the baseline (0.58), which is much better than the best result for our approach was a precision of 0.50. If we increase the minimum number rule occurrences to 100, we are getting the same precision as the baseline but still achieve a higher recall (0.71 to 0.66). Thus the original parameter setting was unsuitable for the English examples.

5.6 Discussion

The results for the 18th century are remarkable, since the precision is increasing from 0.62 to 0.72 for unknown terms and from 0.70 to 0.82 for the spelling variants in the case when one rule is applied and the threshold for the minimum rule occurrences is increased from two to five. A closer look at the evaluation shows that 5 as minimum number of rule occurrences is a very good threshold in this case, since it cuts out a lot of wrong evidences and thus demonstrates the usefulness of the introduced parameters.

Especially the documents from the 16th and 17th centuries contain a lot of unknown terms in foreign languages, mostly Latin. Some documents even contain complete sentences in Latin. In order to avoid showing these terms as unknown terms, a language identifier could be used. This would also offer the possibility of spell-checking the affected passages in the different language.

For insertion and deletion rules taking minimal context into account might further improve the precision. For example, we could replace the insertion rule $\emptyset \rightarrow h$ by rules like $t \rightarrow th$.

Based on the evaluation we must correct our assumption: There are also regularities between spelling variants and the false suggestions. Since some rules are the same as those for spelling variant and the correct suggestion (see Table 1), the false suggestions to some extent even confirm the correct suggestions.

6 Conclusion and Future Work

In this paper, we present a method for automatic construction of evidences. The evidences are needed as input for a rule generation process that enables retrieval for texts in non-standard spelling. The presented approach for automatically creating evidences is very flexible, since the user has several parameters in order to control the process according to his needs with respect to recall and precision of evidences and rules.

The evaluation based on the different parameters showed that the approach for accepting evidences automatically can be applied successfully for creating a training set as well as creating a first set of rules directly. The approach is flexible enough to support different types of user needs. Additional experiments for English show the language-independene of the approach.

The remaining unknown terms will be sorted by decreasing irregularity. We will do that by comparing the n-gram relative frequencies of unknown terms with the corresponding relative n-gram frequencies of a modern collection. We are expecting that the

more frequent terms have a higher probability for containing historic n-grams and thus are good candidates for possible spelling variants.

A user interface with the described approach has already been developed (see [1]). It is integrated into an interactive tool for collecting evidences and a user driven rule generation process where the user can also modify generated rules and create rules on his own (see [8]). At the moment, the automatic evidences are presented in a list of triples consisting of contemporary word, spelling variant and the corresponding rules. Since the evidences are already ordered depending on their rule frequency, we will rearrange the list and group it by rules in order to increase the usability.

Additionally, we will examine if an integration of the Bayes classifier (see Section 2) can enhance the creation of automatic accepted evidences once we have enough examples to train the classifier. In future work, we will also compare the rules that are generated by the automatic evidences to that of the baseline approach.

References

1. Awakian, A.: Development of a user-interface for an interactive rule development. Master thesis, University of Duisburg-Essen (2010)
2. Baron, A., Rayson, P.: VARD 2: A tool for dealing with spelling variation in historical corpora. Proceedings of the Postgraduate Conference in Corpus Linguistics, Aston University, Birmingham (2008)
3. J. Cendrowska: PRISM: An algorithm for inducing modular rules. International Journal of Man-Machine Studies, 27(4), pp. 349-370 (1987)
4. Ernst-Gerlach, A., Fuhr, N.: Generating Search Term Variants for Text Collections with Historic Spellings. In: M. Lalmas, A. MacFarlane, S. Rueger, A. Trombos, T. Tsikrika and A. Yavlinsky (eds): Advances in Information Retrieval - 28th European Conference on IR Research, ECIR 2006. London, UK, April 10-12 2006, Lecture Notes in Computer Science, Vol. 3936, pp.49-60, Springer Verlag, Heidelberg (2006)
5. Ernst-Gerlach, A., Fuhr, N., Retrieval in text collections with historic spelling using linguistic and spelling variants. In: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries, pp. 333-341 Vancouver, BC, Canada ACM, New York, NY, USA (2007)
6. Gotscharek, A., Neumann, A., Reffle, U., Ringlstetter, C., Schulz, K. U.: Enabling Information Retrieval on Historical Document Collections - the Role of Matching Procedures and Special Lexica. In: Proceedings of the ACM SIGIR 2009 Workshop on Analytics for Noisy Unstructured Text Data (AND 2009), pp. 69-76, Barcelona (2009)
7. Hauser, A., Heller, M., Leiss, E., Schulz, K. U., Wanzeck, C.: Information Access to Historical Documents from the Early New High German Period. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-2007) Workshop on Analytics for Noisy Unstructured Text Data, pp. 147-154, Hyderabad, India (2007)
8. Korbar, D.: Visualisation of rule structures and rule modification possibilities for texts with non-standard spelling. Master thesis, University of Duisburg-Essen (2010)
9. Pilz, T.: Nichtstandardisierte Rechtschreibung - Variationsmodellierung und rechnergestützte Variationsverarbeitung. Doctoral thesis, University of Duisburg-Essen (2009)
10. Pilz, T., Luther, W.: Automated support for evidence retrieval in documents with nonstandard orthography. In: The Fruits of Empirical Linguistics, Vol. 1 Process (Sam Featherston, Susanne Winkler eds.) pp. 211-228. Mouton de Gruyter Berlin New York (2009)