Canonicalizing the *deutsches Textarchiv* — DRAFT —

Bryan Jurish
jurish@bbaw.de

1 Introduction

Virtually all conventional text-based natural language processing techniques – from traditional information retrieval systems to full-fledged parsers – require reference to a fixed lexicon accessed by surface form, typically trained from or constructed for synchronic input text adhering strictly to contemporary orthographic conventions. Unconventional input such as historical text which violates these conventions therefore presents difficulties for any such system due to lexical variants present in the input but missing from the application lexicon.

Traditional approaches to the problems arising from an attempt to incorporate historical text into such a system rely on the use of additional specialized (often application-specific) lexical resources to explicitly encode known historical variants. Such specialized lexica are not only costly and time-consuming to create, but also – in their simplest form of static finite word lists – necessarily incomplete in the case of a morphologically productive language like German, since a simple finite lexicon cannot account for highly productive morphological processes such as nominal composition (cf. Kempken et al., 2006).

To facilitate the extension of synchronically-oriented natural language processing techniques to historical text while minimizing the need for specialized lexical resources, one may first attempt an automatic *canonicalization* of the input text. Canonicalization approaches treat orthographic variation phenomena in historical text as instances of an error-correction problem, seeking to map each (unknown) word of the input text to one or more extant *canonical cognates*: synchronically active types which preserve both the root and morphosyntactic features of the associated historical form(s). To the extent that the canonicalization was successful, application-specific process-

ing can then proceed normally using the returned canonical forms as input, without any need for additional modifications to the application lexicon.

This paper provides an informal overview of the various canonicalization techniques currently employed by the *Deutsches Textarchiv*[1] (DTA; Geyken and Klein, 2010) project at the Berlin-Brandenburg Academy of Sciences and Humanities to prepare a corpus of historical German text for part-of-speech tagging, lemmatization, and integration into a robust online information retrieval system. For more details on the methods employed, the interested reader is referred to Jurish (2012).

2 Canonicalization Techniques

It is useful to distinguish between *type-wise* and *token-wise* canonicalization techniques. Type-wise canonicalization techniques are those which process each input word in isolation, independently of its surrounding context, and are fully specified by a binary *conflation relation*[2] over surface strings. Token-wise canonicalization techniques on the other hand make use of the context in which a given instance of a word occurs when determining the optimal canonical cognate, and can thus better account for ambiguities in the mapping from historical to contemporary forms, insofar as these can be resolved by reference to the immediate context. In the sequel, $w \sim_r v$ indicates that the words (types or tokens) w and v are related by the conflator r, and $[w]_r$ denotes the set of all words conflated with w by the conflator r. If r returns a unique (canonical) value v for each input word w, the standard notation for functions $r(w) = v$ will be used.

2.1 String Identity

String identity (id) is the easiest conflator to implement (no additional programming effort or resources are required) and provides a high degree of precision, "false friends" being limited to historical homographs such as the historical form *wider* when it occurs as a variant of the contemporary form *wieder* ("again") rather than the lexically distinct contemporary homograph *wider* ("against"). Since its coverage is restricted to valid contemporary forms, string identity cannot account for any spelling variation at all, resulting in very poor recall – many relevant types will not be retrieved in response to a query in current orthography.

[1] "German Text Archive", `http://www.deutschestextarchiv.de`

[2] Prototypically, every conflation relation will be a true equivalence relation.

As an example, consider the historical form *Abftände*, a variant of the contemporary cognate *Abstände* ("distances"). The conflation set $[Abftände]_{\mathrm{id}} = \{Abftände\}$ does not contain the desired contemporary cognate, so no instances of the historical variant *Abftände* will be retrieved via string identity for a query of the contemporary form *Abstände*. In the DTA canonicalization architecture, string identity is used only as a fallback conflator. Each input word is treated as its own canonical form if all other canonicalizations methods have failed, or if it passes some simple heuristic tests for detecting "uncanonicalizable" strings such as punctuation, abbreviations, mathematical formulæ, or foreign language material in non-latin script.

2.2 Transliteration

A slightly less naïve family of conflation methods are those which employ a simple deterministic transliteration function to replace input characters which do not occur in contemporary orthography with extant equivalents. A transliteration conflator is defined in terms of a character transliteration function xlit which maps each possible input character to a (possibly empty) output string over the contemporary alphabet, the concatenation of which yields the candidate canonical form for the input word.

In the case of historical German, deterministic transliteration is especially useful for its ability to account for typographical phenomena, e.g. by mapping 'ſ' (long 's', as commonly appeared in texts typeset in fraktur) to a conventional round 's', and mapping superscript 'e' to the conventional *Umlaut* diacritic '¨', as in the transliteration $\mathrm{xlit}(Abftände) = Abstände$ ("distances"). Given this transliteration, a query for the contemporary form *Abstände* will successfully retrieve all instances of the historical form *Abftände*.

The DTA canonicalization cascade uses a fast conservative transliteration function based on the `Text::Unidecode` Perl module.[3] Despite its efficiency, and although it outdoes even string identity in terms of its precision, deterministic transliteration suffers from its inability to account for spelling variation phenomena involving extant characters such as the *th/t* and *ey/ei* allographs common in historical German. As an example, consider an instance of the historical form *Theyl* corresponding to the contemporary cognate *Teil* ("part"). Both historical and contemporary forms will be transliterated to themselves, since both strings contain only extant characters, but the historical form will not be retrieved by a query for the contemporary form, since their transliterations are distinct: $\mathrm{xlit}(Teil) = Teil \neq Theyl = \mathrm{xlit}(Theyl)$.

[3] `http://search.cpan.org/~sburke/Text-Unidecode-0.04/`

2.3 Phonetization

A more powerful family of conflation methods is based on the dual intuitions that graphemic forms in historical text were constructed to reflect phonetic forms[4] and that the phonetic system of the target language is diachronically more stable than its graphematic system. A phonetic conflator maps each input word w to a unique phonetic form pho(w) by means of a computable function pho, conflating those strings which share a common phonetic form. The phonetic conversion module used in the DTA was adapted from the phonetization rule-set distributed with the IMS German Festival package (Möhler et al., 2001), a German language module for the Festival text-to-speech system (Black and Taylor, 1997), and compiled as a finite-state transducer.[5]

Phonetic conflation offers a substantial improvement in recall over conservative methods such as transliteration or string identity: variation phenomena such as the *th/t* and *ey/ei* allographs mentioned above are correctly captured by the phonetization transducer: pho(*Theyl*) = [taɪl] = pho(*Teil*), which implies that all instances of the historical form *Theyl* will be retrieved in response to a query of the contemporary form *Teil*. Unfortunately, these improvements come at the expense of precision: in particular, many high-frequency types are misconflated by the simplified phonetization rule-set, including **in* ∼ *ihn* ("in" ∼ "him") and **wider* ∼ *wieder* ("against" ∼ "again"). While such high-frequency cases can easily be dealt with by a small exception lexicon (cf. section 2.6), the underlying tendency of strict phonetic conflation either to over- or to under-generalize – depending on the granularity of the phonetization function – is likely to remain, expressing itself in information retrieval tasks as reduced precision or reduced recall, respectively.

2.4 Rewrite Transduction

Despite its comparatively high recall, the phonetic conflator fails to relate unknown historical forms with any extant equivalent whenever the graphemic variation leads to non-identity of the respective phonetic forms (e.g. pho(*umb*) = [ʔʊmp] ≠ [ʔʊm] = pho(*um*) for the historical variant *umb* of the preposition *um* ("around")), suggesting that recall might be further improved by relaxing the strict identity criterion implicit in the definition of the phonetic conflator. A conflation technique which fulfills both of the above desiderata is *rewrite*

[4]Keller (1978) codifies this intuition as the imperative "write as you speak" governing historical spelling conventions.

[5]In the absence of a language-specific phonetization function, a general-purpose phonetic digest algorithm such as SOUNDEX (Russell, 1918) may be employed instead (Robertson and Willett, 1993).

transduction,[6] which can be understood as a generalization of the well-known *string edit distance* (Damerau, 1964; Levenshtein, 1966).

A rewrite conflator (rw) is defined in terms of a *target lexicon* of contemporary forms and a weighted *error model* (Kernighan et al., 1990; Brill and Moore, 2000) which associates each known pattern of diachronic variation with a non-negative weight or "distance". The conflation set $[w]_{\mathrm{rw},k}$ is computed as the set of k nearest neighbors of the input word w which are themselves members of the target lexicon. Importantly, such a rewrite conflation set can be computed even in the presence of an infinite target lexicon,[7] provided that both lexicon and error model can be represented as (weighted) finite-state transducers (Mohri, 2002).

The DTA canonicalization architecture uses a finite-state rewrite cascade whose error model was compiled from a set of manually constructed rules and whose target lexicon was extracted from the the high-coverage TAGH morphology system for contemporary German (Geyken and Hanneforth, 2006) to compute rewrite conflation sets containing at most only a single "best" contemporary form ($k = 1$). Although this rewrite cascade does indeed improve both precision and recall with respect to the phonetic conflator, these improvements are of comparatively small magnitude, precision in particular remaining well below the level of conservative conflators such as naïve string identity or transliteration, due largely to interference from "false friends" such as the valid contemporary compound *Rockermehl* ("rocker-flour") for the historical variant *Rockermel* of the contemporary form *Rockärmel* ("coat-sleeve").

2.5 Hidden Markov Model Disambiguation

Systematic evaluations of the type-wise techniques described above revealed a typical precision-recall trade-off pattern: the ultra-conservative string identity conflator – despite its near-perfect precision – shows quite poor recall, while the more ambitious high-recall conflators such as phonetic identity or rewrite transduction tend to be disappointingly imprecise. In order to recover some of the precision offered by conservative conflation techniques such as transliteration while still benefiting from the flexibility and improved recall provided by more ambitious techniques such as phonetization or rewrite transduction, the DTA canonicalization architecture makes use of a Hidden Markov Model

[6] Related approaches to historical variant detection include Rayson et al. (2005); Ernst-Gerlach and Fuhr (2006); Gotscharek et al. (2009).

[7] e.g. as arising from morphologically productive phenomena such as German nominal composition

(HMM) disambiguator which operates on the token level, using sentential context to determine a unique "best" canonical form for each input token, in a manner similar to the spelling correction technique described by Mays et al. (1991).

Treating the conflation sets returned by all active type-wise conflators as candidate canonicalization hypotheses, the HMM disambiguator chooses an optimal sequence of token-wise unique canonical forms for each input sentence by application of the well-known Viterbi algorithm (Viterbi, 1967). Lexical probabilities are dynamically computed as a Maxwell-Boltzmann distribution over the candidate conflations for each input word, and a static trigram model of contemporary German is used to model local syntactic and semantic context constraints. The disambiguator is thus able to resolve ambiguous conflation sets such as $\{in, ihn\}$ or $\{wider, wieder\}$ in a context-dependent manner.

Using a simple smoothing mechanism, the disambiguator is also able to override the decisions of the type-wise conflators by selecting a canonical form not explicitly enumerated in the target lexicon.[8] This behavior is particularly useful for proper names, which are not exhaustively represented in the TAGH morphology system, and which were excluded entirely from the rewrite target lexicon because their presence lead to too many spurious conflations with valid historical forms, e.g. the TAGH lexical entry for the surname *Aehnlich* caused the rewrite conflator to treat all instances of the type as their own canonical forms rather than mapping them to the correct contemporary form *ähnlich* ("similar").

2.6 Exception Lexicon

The HMM disambiguator performs very well at the token level, but its reliance on a static n-gram model over contemporary forms is problematic for input words whose canonical cognate was not present in the training data: in such cases, the model effectively reverts to a type-level canonicalization, choosing the most likely conflation candidate based only on criteria of word length and source conflator. Due to the conflator-dependent distance functions employed, short input words in particular are likely to be subjected to such treatment, which was designed primarily to handle low-frequency unlexicalized types such as proper names, and thus often results in a fallback identity

[8] Technically, the possibility of selecting the input word itself as its own canonical form is implemented by allowing the identity conflator id to provide a candidate conflation hypothesis. In practice, the DTA canonicalization architecture uses the transliteration conflator xlit whenever it returns a non-empty string, and only resorts to a pure identity hypothesis when the transliterator fails.

canonicalization. Many common historical variants of high-frequency words fall into this category, usually due to (irregular) patterns of variation not captured by the type-wise conflators such as exhibited by the (strongly inflected) historical variant *frug* of the (weakly inflected) contemporary form *fragte* ("asked"). On the other hand, "false friends" in the training data can cause also spurious canonicalizations: even a single occurrence of the given name *André* in the training data is sufficient to cause the historical variant *andre* of the contemporary form *andere* ("other") to be miscanonicalized.

To handle problematic cases such as these, the DTA canonicalization architecture incorporates a semi-automatically generated exception lexicon (Jurish et al., 2011) which operates on incoming word types before they are passed to the disambiguator. If the exception lexicon contains an entry for an input type, only that entry is considered by the disambiguator as a candidate canonicalization for the input word. This technique ensures that the exception lexicon entry will in fact be the canonical form chosen on the one hand, and allows the disambiguator to make use of the provided entry for context-dependent resolution of nearby items on the other.

3 Summary

Historical text presents unique challenges for typical synchronically oriented natural language processing tasks. In particular, violations of contemporary orthographic conventions are problematic for any task requiring reference to a fixed lexicon keyed by surface word type. Part-of-speech tagging, lemmatization, and information retrieval (corpus indexing & query) are all affected. Canonicalization approaches address this problem by attempting to map unknown historical variants to extant contemporary forms and deferring synchronically oriented analysis to the returned (canonical) forms.

The canonicalization techniques currently used to preprocess the *Deutsches Textarchiv* corpus of historical German were briefly described. String identity on its own does not provide an adequate solution, since it cannot account for any orthographic variation at all, but it can be useful in conjunction with additional heuristics for detecting non-lexical material. Transliteration provides an efficient and very precise canonicalization method for dealing with extinct characters such as the long 's' common in historical German, but cannot account for any variation involving extant characters. More ambitious techniques such as conflation by phonetic identity or rule-based rewrite transduction are able to account for a much wider range of variation, but these improvements come at the cost of precision. Use of a Hidden Markov Model to disambiguate canonicalization hypotheses at the token level using sentential

context effectively recovers much of this lost precision while still benefitting from the improved recall. Remaining systematic canonicalization errors are accounted for by a type-wise exception lexicon. The fully canonicalized corpus was subsequently tagged and lemmatized before being indexed by a robust information retrieval system which uses the canonical-lemma token-level conflation relation to implement an intuitive linguistically motivated search term expansion operator for non-expert user queries.

References

A. W. Black and P. Taylor. Festival speech synthesis system. Technical Report HCRC/TR-83, University of Edinburgh, Centre for Speech Technology Research, 1997. URL `http://www.cstr.ed.ac.uk/projects/festival`.

E. Brill and R. C. Moore. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, 2000.

F. J. Damerau. A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7:171–176, March 1964. doi: 10.1145/363958.363994.

A. Ernst-Gerlach and N. Fuhr. Generating search term variants for text collections with historic spellings. In M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, editors, *Advances in Information Retrieval*, volume 3936 of *Lecture Notes in Computer Science*, pages 49–60. Springer, Berlin, 2006. doi: 10.1007/11735106_6.

A. Geyken and T. Hanneforth. TAGH: A complete morphology for German based on weighted finite state automata. In *Finite State Methods and Natural Language Processing, 5th International Workshop, FSMNLP 2005, Revised Papers*, volume 4002 of *Lecture Notes in Computer Science*, pages 55–66, Berlin, 2006. Springer. doi: 10.1007/11780885_7.

A. Geyken and W. Klein. Deutsches Textarchiv. In *Jahrbuch 2009, Berlin-Brandenburgische Akademie der Wissenschaften*, pages 320–323. Akademie Verlag, Berlin, 2010. URL `http://edoc.bbaw.de/volltexte/2010/1515/pdf/BBAW_Jahrbuch_2009.pdf`.

A. Gotscharek, A. Neumann, U. Reffle, C. Ringlstetter, and K. U. Schulz. Enabling information retrieval on historical document collections: the role of matching procedures and special lexica. In *Proceedings of The Third*

Workshop on Analytics for Noisy Unstructured Text Data, AND '09, pages 69–76, New York, 2009. ACM. doi: 1568296.1568309.

B. Jurish. *Finite-State Canonicalization Techniques for Historical German*. PhD thesis, Universität Potsdam, January 2012. URL `http://opus.kobv.de/ubp/volltexte/2012/5578/`.

B. Jurish, M. Drotschmann, and H. Ast. Constructing a canonicalized corpus of historical German by text alignment. In *Proceedings of the Conference on New Methods in Historical Corpora*, Manchester, UK, 29-30 April 2011. In print.

R. E. Keller. *The German Language*. Faber & Faber, London, 1978.

S. Kempken, W. Luther, and T. Pilz. Comparison of distance measures for historical spelling variants. In M. Bramer, editor, *Artificial Intelligence in Theory and Practice*, pages 295–304. Springer, Boston, 2006. doi: 10.1007/978-0-387-34747-9_31.

M. D. Kernighan, K. W. Church, and W. A. Gale. A spelling correction program based on a noisy channel model. In *Proceedings COLING-1990*, volume 2, pages 205–210, 1990.

V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(1966):707–710, 1966.

E. Mays, F. J. Damerau, and R. L. Mercer. Context based spelling correction. *Information Processing & Management*, 27(5):517–522, 1991. doi: 10.1016/0306-4573(91)90066-U.

G. Möhler, A. Schweitzer, and M. Breitenbücher. *IMS German Festival manual, version 1.2*. Institute for Natural Language Processing, University of Stuttgart, 2001. URL `http://www.ims.uni-stuttgart.de/phonetik/synthesis`.

M. Mohri. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350, 2002.

P. Rayson, D. Archer, and N. Smith. VARD versus Word: A comparsion of the UCREL variant detector and modern spell checkers on English historical corpora. In *Proceedings of the Corpus Linguistics 2005 conference*, Birmingham, UK, July 14-17 2005.

A. M. Robertson and P. Willett. A comparison of spelling-correction methods for the identification of word forms in historical text databases. *Literary and Linguistic Computing*, 8(3):143–152, 1993.

R. C. Russell. Soundex coding system. *United States Patent* 1,261,167, 1918.

A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, pages 260–269, April 1967.