More Than Words: Using Token Context to Improve Canonicalization of Historical German

Bryan Jurish
Berlin-Brandenburgische Akademie der Wissenschaften
jurish@bbaw.de

October 18, 2010

Abstract

Historical text presents numerous challenges for contemporary natural language processing techniques. In particular, the absence of consistent orthographic conventions in historical text presents difficulties for any system requiring reference to a static lexicon indexed by orthographic form. Canonicalization approaches seek to address these issues by associating one or more extant "canonical cognates" with each word of the input text and deferring application analysis to these canonical forms. Type-wise conflation techniques treating each input word in isolation often suffer from a pronounced precision–recall trade-off pattern: high-precision techniques such as conservative transliteration have comparatively poor recall, whereas high-recall techniques such as phonetic conflation tend to be disappointingly imprecise. In this paper, we present a technique for disambiguation of type conflation sets at the token level using a Hidden Markov Model whose lexical probability matrix is dynamically computed from the candidate conflations, and evaluate its performance on a manually annotated corpus of historical German.

Contents

1 Introduction

Historical text presents numerous challenges for contemporary natural language processing techniques. In particular, the absence of consistent orthographic conventions in historical text presents difficulties for any system requiring reference to a fixed lexicon accessed by orthographic form, such as document indexing systems [Sokirko, 2003, Cafarella and Cutting, 2004], part-of-speech taggers [DeRose, 1988, Brill, 1992, Schmid, 1994], simple word stemmers [Lovins, 1968, Porter, 1980], or more sophisticated morphological analyzers [Geyken and Hanneforth, 2006, Zielinski et al., 2009].

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 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.

To facilitate the extension of synchronically-oriented natural language processing techniques to historical text while minimizing the need for specialized lexical resources, we may first attempt an automatic *canonicalization* of the input text. Canonicalization approaches [Jurish, 2008, 2010a] 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 processing can then proceed normally using the returned canonical forms as input, without any need for additional modifications to the application lexicon.

We distinguish between *type-wise* canonicalization techniques which process each input word independently and *token-wise* techniques which make use of the context in which a given instance of a word occurs. In this paper, we present a token-wise canonicalization method which functions as a disambiguator for sets of hypothesized canonical forms as returned by one or more subordinated type-wise techniques. Section 2 provides a brief review of the type-wise canonicalizers used to generate hypotheses, while Section 3 is dedicated to the formal characterization of the disambiguator itself. Section 4 contains a quantitative evaluation of the disambiguator's performance on an information retrieval task over a manually annotated corpus of historical German. Finally, Section 5 provides a brief summary and conclusion.

2 Type-wise Conflation

Type-wise canonicalization techniques are those which process each input word in isolation, independently of its surrounding context. Such a type-wise treatment allows efficient processing of large documents and corpora (since each input type need only be processed once), but disregards potentially useful context information. Formally, a type-wise canonicalization method r is fully specified by a characteristic *conflation relation* $\sim_r$, a binary relation on the set $\mathcal{A}^*$ of all strings over the finite grapheme alphabet $\mathcal{A}$. Prototypically, $\sim_r$ will be a true equivalence relation, inducing a partitioning of the set $\mathcal{A}^*$ of possible word types into equivalence classes or "conflation sets" $[w]_r = \{v \in \mathcal{A}^* : v \sim_R w\}$. In the sequel, we will will use the term "conflation" as synonymous with "type-wise canonicalization", and "conflator" to refer to a specific type-wise canonicalization method.

2.1 String Identity

The simplest of all possible conflators is simple identity of surface strings. The conflation relation $\sim_{\text{id}}$ is in this case nothing more or less than the string identity relation $=$ itself:

$$w \sim_{\text{id}} v :\Leftrightarrow w = v \tag{1}$$

While string identity is the easiest conflator to implement (no additional programming effort or resources are required) and provides a high degree of precision, it cannot account for any graphematic variation at all, resulting in very poor recall. Nonetheless, its inclusion as a conflator ensures that the the set of candidate hypotheses $[w]$ for a given input word w is non-empty,[1] and it provides a baseline with respect to which the relative utility of more sophisticated conflators can be evaluated.

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. Formally, a transliteration conflator is defined in terms of a string transliteration function $\text{xlit} : \mathcal{A}^* \to \widetilde{\mathcal{A}}^*$, where $\mathcal{A}$ is as before a "universal" grapheme alphabet (*e.g.* the set of all Unicode characters) and $\widetilde{\mathcal{A}} \subset \mathcal{A}$ is that subset of the universal alphabet allowed by contemporary orthographic conventions:

$$w \sim_{\text{xlit}} v :\Leftrightarrow \text{xlit}(w) = \text{xlit}(v) \tag{2}$$

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 *Abſt$\overset{\text{e}}{\text{a}}$nde* ↦ *Abstände* ("distances"). For the current work, we used a conservative transliteration function based on the `Text::Unidecode` Perl module[2] Although it rivals raw string identity in terms of its precision, such

[1] $[w]_{\text{id}} \subseteq [w]$ implies $w \in [w]$, and thus $[w] \neq \emptyset$.

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

a conservative transliteration suffers from its inability to account for graphematic variation phenomena involving extant characters such as *th/t* and *ey/ei* alternations common in historical German.

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, and that the phonetic system of the target language is diachronically more stable than its graphematic system. Phonetic conflators map each (historical or extant) word $w \in \mathcal{A}^*$ to a unique phonetic form $\mathrm{pho}(w)$ by means of a computable function $\mathrm{pho} : \mathcal{A}^* \to \mathcal{P}^*$,[3] conflating those strings which share a common phonetic form:

$$w \sim_{\mathrm{pho}} v :\Leftrightarrow \mathrm{pho}(w) = \mathrm{pho}(v) \qquad (3)$$

Note that $[w]_{\mathrm{pho}}$ may be infinite, if for example $\mathrm{pho}(\cdot)$ maps any substring of one or more instances of a single character (e.g. 'a') to a single phon (e.g. /a/). It is useful in such cases to consider the restriction of the conflation set $[w]_{\mathrm{pho}}$ to a finite set of target strings $S \subset \mathcal{A}^*$. We add the superscript "$\upharpoonright S$" to the equivalence class to indicate such a restriction, $[w]^{\upharpoonright S}_{\mathrm{pho}} = [w]_{\mathrm{pho}} \cap S$.

The phonetic conversion module used here 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].[4] Phonetic conflation offers a substantial improvement in recall over conservative methods such as transliteration or string identity. Unfortunately, these improvements often come at the expense of precision.

2.4 Rewrite Transduction

Despite its comparatively high recall, the phonetic conflator fails to relate unknown historical forms with any extant equivalent whenever the graphematic variation leads to non-identity of the respective phonetic forms, suggesting that recall might be further improved by relaxing the strict identity criterion on the right hand side of Equation (3). Moreover, a fine-grained and appropriately parameterized conflator should be less susceptible to precision errors than an "all-or-nothing" (phonetic) identity condition. A technique which fulfills both of the above desiderata is *rewrite transduction*, which can be understood as a generalization of the well-known *string edit distance* [Levenshtein, 1966].

Formally, let $\mathrm{Lex} \subseteq \mathcal{A}^*$ be the (possibly infinite) lexicon of all extant forms, and let Δ_{rw} be a weighted finite-state transducer over a bounded semiring $\mathcal{K}$ which models (potential) diachronic change likelihood as a weighted rational relation. Then define for every input type $w \in \mathcal{A}^*$ the "best" extant equivalent $\mathrm{best}_{\mathrm{rw}}(w)$ as the unique extant type $v \in \mathrm{Lex}$ with minimal edit-distance to the input word:

$$\mathrm{best}_{\mathrm{rw}}(w) = \arg\min_{v \in \mathrm{Lex}} [\![\Delta_{\mathrm{rw}}]\!](w, v) \qquad (4)$$

[3] $\mathcal{P}$ is a finite phonetic alphabet.

[4] In the absence of a language-specific phonetization function, a generic phonetically motivated digest algorithm such as SOUNDEX [Russell and Odell, 1918, Knuth, 1998], the *Kölner Phonetik* [Postel, 1969], or Metaphone [Philips, 1990, 2000] may be employed instead.

Ideally, the image of a word w under best_{rw} will itself be the canonical cognate sought, leading to conflation of all strings which share a common image under best_{rw}:

$$w \sim_{\text{rw}} v :\Leftrightarrow \text{best}_{\text{rw}}(w) = \text{best}_{\text{rw}}(v) \tag{5}$$

For the current experiments, we used the heuristic rewrite transducer described in Jurish [2010a], compiled from 306 manually constructed SPE-style two-level rules, while the target lexicon Lex was extracted from the TAGH morphology transducer [Geyken and Hanneforth, 2006]. Best-path lookup was performed using a specialized variant of the well-known *Dijkstra algorithm* [Dijkstra, 1959] as described in Jurish [2010b]. 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.

3 Token-wise Disambiguation

In an effort to recover some degree 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, we have developed a method for disambiguating type-wise conflation sets which operates on the token level, using sentential context to determine a unique "best" canonical form for each input token. Specifically, the disambiguator employs a Hidden Markov Model (HMM) whose lexical probability matrix is dynamically re-computed for each input sentence from the conflation sets returned by one or more subordinated type-wise conflators, and whose transition probabilities are given by a static word n-gram model of the target language, *i.e.* present-day German adhering to current orthographic conventions.

3.1 Basic Model

Formally, let $\mathcal{W} \subset \widetilde{\mathcal{A}}^*$ be a finite set of known extant words, let $\mathbf{u} \notin \mathcal{W}$ be a designated symbol representing an unknown word, let $S = \langle w_1, \ldots, w_{n_S} \rangle$ be an input sentence of n_S (historical) words with $w_i \in \mathcal{A}^*$ for $1 \leq i \leq n_S$, and let $R = \{r_1, \ldots, r_{n_R}\}$ be a finite set of (opaque) type-wise conflators. Then, the disambiguator HMM is defined in the usual way [Rabiner, 1989, Charniak et al., 1993, Manning and Schütze, 1999] as the 5-tuple $D = \langle \mathcal{Q}, \mathcal{O}_S, \Pi, A, B_S \rangle$, where:

1. $\mathcal{Q} = (\mathcal{W} \cup \{\mathbf{u}\}) \times R$ is a finite set of model *states*, where each state $q \in \mathcal{Q}$ is pair $\langle \tilde{w}_q, r_q \rangle$ composed of an extant word form $\tilde{w}_q$ and a conflator r_q;
2. $\mathcal{O}_S = \text{rng}(S) = \bigcup_{i=1}^{n_S} \{w_i\}$ is the set of *observations* for the input sentence S;
3. $\Pi : \mathcal{Q} \to [0,1] : q \mapsto p(Q_1 = q)$ is a static probability distribution over $\mathcal{Q}$ representing the model's *initial state probabilities*;
4. $A : \mathcal{Q}^k \to [0,1] : \langle q_1, \ldots, q_k \rangle \mapsto p(Q_i = q_k | Q_{i-k+1} = q_1, \ldots, Q_{i-1} = q_{k-1})$ is a static conditional probability distribution over $\mathcal{Q}$ k-grams representing the model's *state transition probabilities*; and

5. $B_S : \mathcal{Q} \times \mathcal{O}_S \to [0,1] : \langle q, o\rangle \mapsto p(O = o|Q = q)$ is a dynamic probability distribution over observations conditioned on states representing the model's *lexical probabilities.*

3.2 Transition Probabilities

The finite target lexicon $\mathcal{W}$ can easily be extracted from a corpus of contemporary text. For estimating the static distributions Π and A, we first make the following assumptions:

$$p(Q = \langle \tilde{w}_q, r_q\rangle) = p(W = \tilde{w}_q)p(R = r_q) \tag{6}$$

$$p(R = r) = \frac{1}{n_R} \tag{7}$$

Equation 6 asserts the independence of extant forms and conflators, while Equation 7 assumes a uniform distribution over conflators. Given these assumptions, the static state distributions Π and A can be estimated as:

$$\Pi(q) :\approx p\left(W_1 = \tilde{w}_q\right)/n_R \tag{8}$$

$$A(q_1, \ldots, q_k) :\approx p\left(W_i = \tilde{w}_{q_k} | W_{i-k+1}^{i-1} = \tilde{w}_{q_1} \ldots \tilde{w}_{q_{k-1}}\right)/n_R \tag{9}$$

Equations (8) and (9) are nothing more or less than a word k-gram model over extant forms, scaled by the constant $\frac{1}{n_R}$. We can therefore use standard maximum likelihood techniques to estimate Π and A from a corpus of contemporary text [Bahl et al., 1983, Manning and Schütze, 1999].

For the current experiments, we trained a word trigram model ($k = 3$) on the TIGER corpus of contemporary German [Brants et al., 2002]. Probabilities for the "unknown" form $\mathbf{u}$ were computed using the simple smoothing technique of assigning $\mathbf{u}$ a pseudo-frequency of $\frac{1}{2}$ [Lidstone, 1920, Manning and Schütze, 1999]. To account for unseen trigrams, the resulting trigram model was smoothed by linear interpolation of uni-, bi-, and trigrams [Jelinek and Mercer, 1980, 1985], using the method described by Brants [2000] to estimate the interpolation coefficients.

3.3 Lexical Probabilities

In the absence of a representative corpus of conflator-specific manually annotated training data, we cannot use maximum likelihood techniques to estimate the model's lexical probabilities B_S. Instead, lexical probabilities are instantiated as a Maxwell-Boltzmann distribution:

$$B\big(\langle \tilde{w}, r\rangle, w\big) :\approx \frac{b^{\beta d_r(w,\tilde{w})}}{\sum_{r' \in R} \sum_{\tilde{w}' \in [w]_{r'}} b^{\beta d_{r'}(w,\tilde{w}')}} \tag{10}$$

Here, $b, \beta \in \mathbb{R}$ are free model parameters with $\beta < 0 < b$, and for a conflator $r \in R$, the function $d_r : \mathcal{A}^* \times \mathcal{W} \to \mathbb{R}_+$ is a pseudo-metric used to estimate the reliability of the conflator's association of an input word w with the extant form $\tilde{w}$

It should be explicitly noted that the denominator of the right-hand side of Equation (10) is a sum over all model states (canonicalization hypotheses) $\langle \tilde{w}', r'\rangle$ actually associated with the observation argument w by the type-wise conflation stage, and *not* a sum over observations w' associable with the state argument $\langle \tilde{w}, r \rangle$. This latter sum (if it could be computed) would adhere to the traditional form $\left(\mathrm{sim}(o,q)/\sum_{o'} \mathrm{sim}(o',q)\right)$ for estimating a probability distribution $p(O|Q)$ over *observations* conditioned on model states such as the HMM lexical probability matrix B_S is defined to represent; whereas the estimator in Equation (10) is of the form $\left(\mathrm{sim}(o,q)/\sum_{q'} \mathrm{sim}(o,q')\right)$, which corresponds more closely to a distribution $p(Q|O)$ over *states* conditioned on observations.[5]

From a practical standpoint, it should be clear that Equation (10) is much more efficient to compute than an estimator summing globally over potential observations, since all the data needed to compute Equation (10) are provided by the type-wise preprocessing of the input sentence S itself, whereas a theoretically pure global estimator would require a whole arsenal of *inverse* conflators as well as a mechanism for restricting their outputs to some tractable set of admissable historical forms, and hence would be of little practical use. From a formal standpoint, we believe that our estimator as used in the run-time disambiguator can be shown to be equivalent to a global estimator, provided that the conflator pseudo-metrics d_r are symmetric and the languages of both historical and extant forms are uniformly dense, but a proof of this conjecture is beyond the scope of the current work.

It was noted above in Section 2.3 that the for the phonetic conflator in particular, the equivalence class $[w]_{\mathrm{pho}} = \{v \in \mathcal{A}^* : w \sim_{\mathrm{pho}} v\}$ may not be finite. In order to ensure the computational tractability of Equation (10) therefore, the phonetic conflations considered were implicitly restricted to the finite set $\mathcal{W}$ of known extant forms used to define the model's states, $[w]_{\mathrm{pho}}^{\restriction \mathcal{W}}$. Transliterations and rewrite targets which were not also known extant forms were implicitly mapped to the designated symbol $\mathbf{u}$ for purposes of estimating transition probabilities for previously unseen extant word types.

For the current experiments, we used the following model parameters:

$$\begin{array}{rcll}
b & = & 2 & \\
\beta & = & -1 & \\
d_{\mathrm{xlit}}(w,\tilde{w}) & = & 2/|w| & \text{if } \tilde{w} = \mathrm{xlit}(w) \\
d_{\mathrm{pho}}(w,\tilde{w}) & = & 1/|w| & \text{if } \tilde{w} \in [w]_{\mathrm{pho}}^{\restriction \mathcal{W}} \\
d_{\mathrm{rw}}(w,\tilde{w}) & = & [\![\Delta_{\mathrm{rw}}]\!](w,\tilde{w})/|w| & \text{if } \tilde{w} \in [w]_{\mathrm{rw}}
\end{array}$$

In all other cases, $d_r(w,\tilde{w})$ is undefined. Note that all conflator distance functions are scaled by inverse input word length $\frac{1}{|w|}$. Defining distance functions in terms of (inverse) word length in this manner captures the intuition that a conflator is less likely to discover a false positive conflation for a longer input word than for a short one; natural language lexica tending to be maximally dense for short (usually closed-class) words. The transliteration and phonetic conflators are constants given input word length, whereas the rewrite conflator makes use of the cost $[\![\Delta_{\mathrm{rw}}]\!](w,\tilde{w})$ assigned to the conflation pair by the rewrite FST itself.

3.4 Runtime Disambiguation

Having defined the disambiguator model, we can use it to determine a unique "best" canonical form for each input sentence S by applying the well-known

[5] See the discussion surrounding Equation 20 in Charniak et al. [1993] for a more detailed look at these two sorts of lexical probability estimator and their effects on HMM part-of-speech taggers.

Viterbi algorithm [Viterbi, 1967]. Formally, the Viterbi algorithm computes the state path with maximal probability given the observed sentence:

$$\textsc{Viterbi}(S, D) = \vec{Q} = \operatorname*{arg\,max}_{\langle q_1,\ldots,q_{n_S}\rangle \in \mathcal{Q}^{n_S}} p(q_1, \ldots, q_{n_S}|S, D) \tag{11}$$

Finally, extracting the disambiguated canonical forms from the state sequence $\vec{Q}$ returned by the Viterbi algorithm is a trivial matter of projecting the extant form components of the HMM state structures:

$$\textsc{Disambig}(S, D) = \langle \tilde{w}_{\vec{Q}(1)}, \ldots, \tilde{w}_{\vec{Q}(n_S)} \rangle \tag{12}$$

4 Evaluation

4.1 Test Corpus

The conflation and disambiguation techniques described above were tested on a manually annotated corpus of historical German. The test corpus was comprised of the full body text from 13 volumes published between 1780 and 1880, and contained 152,776 tokens of 17,417 distinct types in 9,079 sentences, discounting non-alphabetic types such as punctuation. To assign an extant canonical equivalent to each token of the test corpus, the text of each volume was automatically aligned token-wise with a contemporary edition of the same volume. Automatically discovered non-identity alignment pair types were presented to a human annotator for confirmation. In a second annotation pass, all tokens lacking an identical or manually confirmed alignment target were inspected in context and manually assigned a canonical form. Whenever they were presented to a user, proper names and extinct lexemes were treated as their own canonical forms. In all other cases, equivalence was determined by direct etymological relation of the root in addition to matching morphosyntactic features. Problematic tokens were marked as such and subjected to expert review. Marginalia, front and back matter, speaker and stage directions, and tokenization errors were excluded from the final evaluation corpus.

4.2 Evaluation Measures

The canonicalization methods from Sections 2 and 3 were evaluated using the gold-standard test corpus to simulate a document indexing and query scenario. Formally, let $C = \{c_1, \ldots, c_{n_C}\}$ be a finite set of canonicalizers, and let $G = \langle S_1, \ldots, S_{n_G} \rangle$ represent the sentences of the test corpus, where each sentence $S_i = \langle g_{i;1}, \ldots, g_{i;n_{S_i}} \rangle$ is a string of token-tuples $g_{i;j} = \langle w_{i;j}, \tilde{w}_{i;j}, [w_{i;j}]_{c_1}, \ldots, [w_{i;j}]_{c_{n_C}} \rangle$, $1 \leq i \leq n_G$ and $1 \leq j \leq n_{S_i}$. Here, $w_{i;j}$ represents the literal token text as appearing in the historical corpus, $\tilde{w}_{i;j}$ is its gold-standard canonical cognate, and $[w_{i;j}]_{c_k}$ represents the set of canonical form(s) assigned to the token by the canonicalizer c_k. Let $Q = \bigcup_{i=1}^{n_G} \bigcup_{j=1}^{n_{S_i}} \{\tilde{w}_{i;j}\}$ be the set of all canonical cognates represented in the corpus, and define for each canonicalizer $c \in C$ and query string $q \in Q$ the sets $\text{relevant}(q), \text{retrieved}_c(q) \subseteq \mathbb{N}^2$ of *relevant* and *retrieved* corpus tokens as:

$$\text{relevant}(q) = \{\langle i, j \rangle \in \mathbb{N}^2 : q = \tilde{w}_{i;j}\} \tag{13}$$

$$\text{retrieved}_c(q) = \{\langle i, j \rangle \in \mathbb{N}^2 : q \in [w_{i;j}]_c\} \tag{14}$$

	% Types			**% Tokens**		
c	$\mathrm{pr}_{\mathrm{typ}}$	$\mathrm{rc}_{\mathrm{typ}}$	$\mathrm{F}_{\mathrm{typ}}$	$\mathrm{pr}_{\mathrm{tok}}$	$\mathrm{rc}_{\mathrm{tok}}$	$\mathrm{F}_{\mathrm{tok}}$
id	99.0	59.2	74.1	99.8	79.3	88.4
xlit	**99.1**	89.5	94.1	**99.8**	96.8	98.3
pho	97.1	96.1	96.6	91.4	99.2	95.1
rw	97.6	**96.5**	**97.0**	94.3	**99.3**	96.7
hmm	98.6	95.3	96.9	99.7	99.1	**99.4**

Table 1: Evaluation data for various canonicalization techniques

Token-wise precision and recall for the canonicalizer c can then be defined as:

$$\mathrm{pr}_{\mathrm{tok}} = \frac{\left|\bigcup_{q \in Q} \mathrm{retrieved}_c(q) \cap \mathrm{relevant}(q)\right|}{\left|\bigcup_{q \in Q} \mathrm{retrieved}_c(q)\right|} \tag{15}$$

$$\mathrm{rc}_{\mathrm{tok}} = \frac{\left|\bigcup_{q \in Q} \mathrm{retrieved}_R(q) \cap \mathrm{relevant}(q)\right|}{\left|\bigcup_{q \in Q} \mathrm{relevant}(q)\right|} \tag{16}$$

Type-wise measures are defined analogously, by mapping the token index sets of Equations (13) and (14) to corpus types before applying Equations (15) and (16). We use the unweighted harmonic precision-recall average F [van Rijsbergen, 1979] as a composite measure for both type- and token-wise evaluation modes:

$$\mathrm{F}(pr, rc) = \frac{2 \cdot pr \cdot rc}{pr + rc} \tag{17}$$

4.3 Results

Qualitative results for the canonicalization techniques described in Sections 2 and 3 with respect to the test corpus are given in Table 1. Immediately apparent from the data is the typical precision–recall trade-off pattern discussed above: conservative conflators such as string identity (id) and transliteration (xlit) have near-perfect precision ($\geq 99\%$ both type- and token-wise), but relatively poor recall. On the other hand, ambitious conflators such as phonetic identity (pho) or the heuristic rewrite transducer (rw) reduce type-wise recall errors by over 66% and token-wise recall errors by over 75%, with respect to transliteration, but these recall gains come at the expense of precision.

As hoped, the HMM disambiguator (hmm) presented in Section 3 does indeed recover a large degree of the precision lost by the ambitious type-wise conflators, achieving a reduction of over 41% in type-wise precision errors and over 94% in token-wise precision errors with respect to the heuristic rewrite conflator. While some additional recall errors are made by the HMM, there are comparatively few of these, so that the harmonic average F falls by a mere 3% with respect to the highest-recall method (rw). Indeed, the token-wise composite measure F is substantially higher for the HMM disambiguator (99.4%, *versus* 96.7% for the rewrite method), outperforming its closest competitor — deterministic transliteration (xlit) — by over 64%.

The most surprising aspect of these results is the recall performance of the conservative transliterator xlit with $\mathrm{rc}_{\mathrm{tok}} = 96.8\%$. While such performance combined with the ease of implementation and computational efficiency of the transliteration method makes it very attractive at first glance, note that the test corpus was drawn from a comparatively recent text sample, and that a diachronically more heterogeneous corpus such as that described in Jurish [2010a] is likely to be less amenable to such simple techniques.

5 Conclusion

We have identified a typical precision–recall trade-off pattern exhibited by several type-wise conflation techniques used to automatically discover extant canonical forms for historical German text. Conservative conflators such as string identity and transliteration return very precise results, but suffer from comparatively poor recall. More ambitious techniques such as conflation by phonetic form or heuristic rewrite transduction show a marked improvement in recall, but disappointingly poor precision. To address these problems, we proposed a method for disambiguating type conflation sets at the token level using sentential context to optimize the path probability of canonical forms conditioned on observed historical forms. The disambiguator uses a Hidden Markov Model whose lexical probabilities are dynamically re-computed for every input sentence based on the conflation hypotheses returned by a set of subordinated type-wise conflators.

The proposed disambiguation architecture was evaluated on an information retrieval task over a gold standard corpus of manually confirmed canonicalizations of historical German text. Use of the token-wise disambiguator provided a precision error reduction of over 94% with respect to the best recall method, and a recall error reduction of over 71% with respect to the most precise method. Overall, the proposed disambiguation method performed best at the token level, achieving a token-wise F of 99.4%.

We are interested in verifying our results using larger and less homogeneous corpora than the test corpus used here, as well as extending the techniques described here to other languages and domains. In future work, we wish to implement and test a language-independent type-wise conflator such as that described by Kondrak [2000], and to systematically investigate the effects of the various disambiguator parameters as well as more sophisticated smoothing techniques for handling previously unseen extant types and sparse training data.

References

L. R. Bahl, F. Jelinek, and R. L. Mercer. A Maximum Likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Pami-5(2):179–190, 1983.

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`.

S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, 2002.

T. Brants. TnT – a statistical part-of-speech tagger. In *Proceedings of ANLP-2000*, 2000.

E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, Italy, 1992.

M. Cafarella and D. Cutting. Building Nutch: Open source search. *Queue*, 2(2):54–61, 2004. ISSN 1542-7730. doi: http://doi.acm.org/10.1145/988392.988408.

E. Charniak, C. Hendrickson, N. Jacobson, and M. Perkowitz. Equations for part-of-speech tagging. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 784–789, 1993.

S. DeRose. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14(1):31–39, 1988.

E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

A. Geyken and T. Hanneforth. TAGH: A complete morphology for German based on weighted finite state automata. In *Proceedings FSMNLP 2005*, pages 55–66, Berlin, 2006. Springer. doi: http://dx.doi.org/10.1007/11780885_7.

F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice*, pages 381–397. North-Holland Publishing Company, Amsterdam, 1980.

F. Jelinek and R. L. Mercer. Probability distribution estimation from sparse data. *IBM Technical Disclosure Bulletin*, 28:2591–2594, 1985.

B. Jurish. Finding canonical forms for historical German text. In A. Storrer, A. Geyken, A. Siebert, and K.-M. Würzner, editors, *Text Resources and Lexical Knowledge*, pages 27–37. Mouton de Gruyter, Berlin, 2008. ISBN 978-3-11-020735-4.

B. Jurish. Comparing canonicalizations of historical German text. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 72–77, Uppsala, Sweden, July 2010a. URL `http://www.aclweb.org/anthology/W10-2209`.

B. Jurish. Efficient online k-best lookup in weighted finite-state cascades. In T. Hanneforth and G. Fanselow, editors, *Language and Logos: Studies in Theoretical and Computational Linguistics*, volume 72 of *Studia grammatica*. Akademie Verlag, Berlin, 2010b. ISBN 978-3-05-004931-1.

D. Knuth. *The Art of Computer Programming, Volume 3: Sorting And Searching. Second Edition.* Addison-Wesley, Reading, MA, 1998. ISBN 0-201-89685-0.

G. Kondrak. A new algorithm for the alignment of phonetic sequences. In *Proceedings NAACL*, pages 288–295, 2000.

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

G. J. Lidstone. Note on the general case of the Bayes-Laplace formula for inductive or *a priori* probabilities. *Transactions of the Faculty of Actuaries*, 8:182–192, 1920.

J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

C. D. Manning and H. Schütze. *Foundations of statistical natural language processing.* MIT Press, Cambridge, MA, 1999.

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`.

L. Philips. Hanging on the metaphone. *Computer Language*, 7(12):39, December 1990.

L. Philips. The double metaphone search algorithm. *C/C++ Users Journal*, June 2000, June 2000.

M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

H. J. Postel. Die Kölner Phonetik. Ein Verfahren zur Identifizierung von Personennamen auf der Grundlage der Gestaltanalyse. *IBM-Nachrichten*, 19:925–931, 1969.

L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

R. C. Russell and M. K. Odell. Soundex phonetic coding system. *US Patent* 1,261,167, 1918.

H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK, 1994.

A. Sokirko. A technical overview of DWDS/dialing concordance. Talk delivered at the meeting *Computational linguistics and intellectual technologies*, Protvino, Russia, 2003. URL `http://www.aot.ru/docs/OverviewOfConcordance.htm`.

C. J. van Rijsbergen. *Information Retrieval.* Butterworth-Heinemann, Newton, MA, 1979. ISBN 0408709294.

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

A. Zielinski, C. Simon, and T. Wittl. Morphisto: Service-oriented open source morphology for German. In C. Mahlow and M. Piotrowski, editors, *State of the Art in Computational Morphology*, volume 41 of *Communications in Computer and Information Science*, pages 64–75. Springer, Berlin, 2009. ISBN 978-3-642-04131-0. URL `http://dx.doi.org/10.1007/978-3-642-04131-0_5`.