Bryan Jurish

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

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 information retrieval 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).[1]

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 (Jurish, 2008, 2010a; Gotscharek et al., 2009a) treat orthographic variation phenomena in historical text as instances of an error-correction problem (Shannon, 1948; Kukich, 1992; Brill and Moore, 2000), 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.

I 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, I present a token-wise canonicalization

[1] While neither information retrieval (IR) systems nor stemmers use a *static* fixed lexicon in the usual sense, the effective lexicon of an IR system is fixed at indexing time as the set of all actually occurring word forms. Similarly, the lexicon of a traditional stemmer has a static portion (hard-coded inflection rules) as well as a dynamic portion (set of stems) determined by the actual input. In both cases, historical spelling variants will be treated as distinct lexemes rather than associated with an equivalent contemporary cognate unless additional measures such as those described here are taken.

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 conflation 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 conflator 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" $[w]_r = \{v \in \mathcal{A}^* : v \sim_r w\}$ induced by some type $w \in \mathcal{A}^*$. Where appropriate, I distinguish between the full conflation set $[w]_r$ containing all strings conflated by r with w and a conflator-specific finite subset $\downarrow[w]_r \subseteq [w]_r$ representing the *canonicalization hypotheses* provided by r for w: the former sets will be used to characterize the retrieval function for r used to define the evaluation measures precision and recall in section 4.2, while the latter will be used in the definition of the token-wise disambiguator in section 3.3. Unless otherwise specified, I assume $\downarrow[w]_r = [w]_r$. In the sequel, I will use the terms "conflation" and "type-wise canonicalization" interchangeably where no ambiguity will result, and the term "conflator" will be used to refer to a specific type-wise canonicalization method.

2.1 String Identity

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

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

String identity 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 are not retrieved in response to a query in current orthography. Nonetheless, its inclusion as a conflator ensures that the

set of candidate hypotheses $[w]$ for a given input word w is non-empty,[2] and it provides a baseline with respect to which the relative utility of more sophisticated conflators can be evaluated.

As an example, consider the historical form *Abftände*, a variant of the contemporary cognate *Abstände* ("distances"). The conflation set $[\textit{Abftände}]_{\text{id}} = \{\textit{Abftände}\}$ is non-empty, but does not contain the desired contemporary cognate ($\textit{Abstände} \notin [\textit{Abftände}]_{\text{id}}$), so Equation (20) from section 4.2 dictates that no instances of the historical variant *Abftände* will be retrieved via string identity for a query of the contemporary form *Abstände*.

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 character 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[3] characters) and $\widetilde{\mathcal{A}} \subseteq \mathcal{A}$ is that subset of the universal alphabet allowed by contemporary orthographic conventions. The elementary character transliteration function is extended to a string transliteration function $\text{xlit}^* : \mathcal{A}^* \to \widetilde{\mathcal{A}}^*$ in the usual manner by iteratively applying xlit to each character of the input string in turn (Equation 2), canonicalization hypotheses are limited to the transliterator output (Equation 3), and the characteristic conflation relation $\sim_{\text{xlit}}$ is defined as identity of transliterated strings (Equation 4):

$$\text{xlit}^*(a_1 a_2 \ldots a_n) := \text{xlit}(a_1)\,\text{xlit}(a_2)\ldots\text{xlit}(a_n) \quad (2)$$

$$\downarrow[w]_{\text{xlit}} := \{\text{xlit}^*(w)\} \quad (3)$$

$$w \sim_{\text{xlit}} v :\Leftrightarrow \text{xlit}^*(w) = \text{xlit}^*(v) \quad (4)$$

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 *Abftände* ↦ *Abstände* ("distances"). Given this transliteration, a query for the contemporary form *Abstände* will successfully retrieve all instances of the historical form *Abftände*: $\text{xlit}^*(\textit{Abstände}) = \textit{Abstände} = \text{xlit}^*(\textit{Abftände})$, so $\textit{Abstände} \in [\textit{Abftände}]_{\text{xlit}}$.

The current work makes use of a conservative transliteration function based on the `Text::Unidecode` Perl module.[4] Due to the fact that the underlying character transliteration table is comparatively small and can be implemented as an in-memory

[2] Since $[w]_{\text{id}} = \{w\}$, $[w]_{\text{id}} \subseteq [w]$ implies $w \in [w]$, and thus $[w] \neq \emptyset$. Since the more reliable transliterating conflator described in section 2.2 also ensures a non-empty set of conflation hypotheses, the identity conflator itself was not used to generate hypotheses for the disambiguator in the current experiments.

[3] Unicode Consortium (2011), `http://www.unicode.org/`

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

array, transliteration is a very efficient conflation method, with $\mathcal{O}(\text{xlit}) = \mathcal{O}(1)$ and therefore $\mathcal{O}(\text{xlit}^*) = \mathcal{O}(n)$. In terms of expressive power, since xlit is finite, it can be represented by a finite state transducer, and therefore so can its reflexive and transitive closure xlit^*.

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: $\text{xlit}^*(Teil) = Teil \neq Theyl = \text{xlit}^*(Theyl)$ implies $Teil \not\sim_{\text{xlit}} Theyl$ and therefore $Teil \notin [Theyl]_{\text{xlit}}$.

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[5] 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 $\text{pho}(w)$ by means of a computable function $\text{pho} : \mathcal{A}^* \to \mathcal{P}^*$,[6] conflating those strings which share a common phonetic form:

$$w \sim_{\text{pho}} v :\Leftrightarrow \text{pho}(w) = \text{pho}(v) \tag{5}$$

Since $[w]_{\text{pho}}$ may be infinite – if for example $\text{pho}(\cdot)$ maps any substring of one or more instances of a single character (e.g. 'a') to a single phone (e.g. [a]) – additional care must be taken to ensure a finite set of canonicalization hypotheses $\downarrow[w]_{\text{pho}}$. A straightforward way to ensure a finite hypothesis set is simply to restrict $[w]_{\text{pho}}$ to some finite set of pre-defined target strings $T \subset \mathcal{A}^*$, setting $\downarrow[w]_{\text{pho}} = \downarrow_T[w]_{\text{pho}} = [w]_{\text{pho}} \cap T$. If pho can be represented as a finite-state transducer M_{pho} and the target lexicon can be represented as a finite-state acceptor A_{Lex}, a more robust alternative is to use a k-best string lookup algorithm such as that described in Jurish (2010b) on the cascade $\mathcal{C}_{\text{pho}}(w) = \text{Id}(w) \circ M_{\text{pho}} \circ M_{\text{pho}}^{-1} \circ A_{\text{Lex}}$, defining $\downarrow[w]_{\text{pho}} = \downarrow_{\mathcal{C},k}[w]_{\text{pho}} = \text{kbest}(k, \mathcal{C}_{\text{pho}}(w))$ for some finite upper bound k on the number of admissible hypotheses, assuming an appropriate weighting scheme on A_{Lex}.

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)

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

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

and compiled as a finite-state transducer (Jurish, 2008).[7] 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 $Teil \in [Theyl]_{\text{pho}}$. Unfortunately, these improvements often 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"), **statt* ∼ *Stadt*, ("instead" ∼ "city"), and **wider* ∼ *wieder* ("against" ∼ "again"). While such high-frequency cases might be easily handled in a mature system by a small exception lexicon, 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 on the right hand side of Equation (5). Moreover, a fine-grained and appropriately parameterized conflator should be less susceptible to precision errors than an "all-or-nothing" (phonetic) identity condition (Kondrak, 2000, 2002). 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* (Damerau, 1964; Levenshtein, 1966).

Formally, let $\text{Lex} \subseteq \mathcal{A}^*$ be the (possibly infinite) lexicon of all extant forms encoded as a finite-state acceptor A_{Lex}, and let M_{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 $\text{best}_{\text{rw}}(w)$ as the unique extant type $v \in \text{Lex}$ with minimal edit-distance to the input word:

$$\text{best}_{\text{rw}}(w) = \arg\min_{v \in \mathcal{A}^*} [\![M_{\text{rw}} \circ A_{\text{Lex}}]\!](w, v) \tag{6}$$

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}:

$$w \sim_{\text{rw}} v :\Leftrightarrow \text{best}_{\text{rw}}(w) = \text{best}_{\text{rw}}(v) \tag{7}$$

[7] In the absence of a language-specific phonetization function, a general-purpose phonetic digest algorithm such as SOUNDEX (Russell, 1918), the *Kölner Phonetik* (Postel, 1969), PHONIX (Gadd, 1988, 1990), or Metaphone (Philips, 1990, 2000) may be employed instead (Robertson and Willett, 1993; Kempken, 2005).

The current experiments were performed using the heuristic rewrite transducer described in Jurish (2010a), compiled from 306 manually constructed two-level rules, while the lexical target acceptor A_{Lex} was extracted from the TAGH morphology transducer (Geyken and Hanneforth, 2006). The native TAGH weights were scaled for compatibility and used to provide a prior cost distribution over target word forms based on their derivational complexity. Best-path lookup was performed using a specialized variant of the well-known *Dijkstra algorithm* (Dijkstra, 1959) as described in Jurish (2010b). Related approaches to historical variant detection include Kempken (2005); Rayson et al. (2005); Ernst-Gerlach and Fuhr (2006); Gotscharek et al. (2009a).

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") as appearing in Figure 1.

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, I 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 k-gram model of the target language, in this case contemporary German adhering to current orthographic conventions. Similar approaches for traditional spell-checking applications using strictly local context for language modelling have been described by Kernighan et al. (1990); Church and Gale (1991); Brill and Moore (2000); Verberne (2002). Most closely related to the current proposal is the approach of Mays et al. (1991), who use a word trigram model to disambiguate unweighted confusion sets returned by a traditional approximate Damerau-Levenshtein matcher analogous to the rewrite cascade from section 2.4. An example of the proposed disambiguation architecture for the conflators described in section 2 is given in Figure 1.

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

	Dete	ſammlete	Steyne	im	Rockermel
id	Dete	*fammlete*	*Steyne*	im	*Rockermel*
xlit	Dete	*sammlete*	*Steyne*	im	*Rockermel*
pho	∅	∅	{Steine}	{im, *ihm*}	{Rockärmel}
rw	*Tete*⟨1⟩	sammelte⟨5⟩	Steine⟨1⟩	im⟨0⟩	*Rockermehl*⟨10⟩
hmm	*Dete*	*sammelte*	*Steine*	*im*	*Rockärmel*

Figure 1: Example of the proposed conflator disambiguation architecture for the input sentence *"Dete ſammlete Steyne im Rockermel"* ("Dete gathered rocks in the coat-sleeve"). Costs assigned by the rewrite transducer appear in angled brackets, and the conflation hypotheses selected by the HMM disambiguator are underlined.

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 a pair $\langle \tilde{w}_q, r_q \rangle$ composed of an extant word form $\tilde{w}_q$ and a conflator r_q;
2. $\mathcal{O}_S = \bigcup_{i=1}^{n_S} \{w_i\}$ 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 state 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*.

Using the shorthand notation w_i^{i+j} for the string $w_i w_{i+1} \ldots w_{i+j}$, the model D computes sentential probability as the sum of path probabilities over all possible generating state sequences:

$$p(S = w_1^{n_S}) \quad = \quad \sum_{q_1^{n_S} \in \mathcal{Q}^{n_S}} p(S = w_1^{n_S}, Q = q_1^{n_S}) \tag{8}$$

Assuming suitable boundary handling for negative indices, joint path probabilities themselves are computed as:

$$p(S = w_1^{n_S}, Q = q_1^{n_S}) \quad = \quad \prod_{i=1}^{n_S} p(q_i | q_{i-k+1}^{i-1}) p(w_i | q_i) \tag{9}$$

Underlying these equations are the following Markov assumptions:

$$p(q_i|q_1^{i-1}, w_1^{i-1}) = p(q_i|q_{i-k+1}^{i-1}) \quad (10)$$

$$p(w_i|q_1^{i}, w_1^{i-1}) = p(w_i|q_i) \quad (11)$$

Equation (10) asserts that state transition probabilities depend on at most the preceding $k-1$ states. Equation (11) asserts the independence of observed surface forms (historical spellings) from all but the model's current state. Taken together, these assumptions will lead to the use of a k-gram distribution over contemporary word forms to model both syntactic and (local) semantic constraints of the target language as operating on conflator-dependent type-wise canonicalization hypotheses for historical input forms. Crucially, the product of these two component distributions as used in the path probability computation from Equation (9) will allow linguistic context constraints (insofar as they are captured by the k-gram transition probabilities) to override prior type-wise estimates of a conflation's reliability (and vice versa), leading to a disambiguator dependent on both token context and prior estimates of conflation likelihood.

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:

$$p(Q = \langle \tilde{w}_q, r_q \rangle) = p(W = \tilde{w}_q)p(R = r_q) \quad (12)$$

$$p(R = r) = \frac{1}{n_R} \quad (13)$$

Equation (12) asserts the independence of extant forms and conflators, while Equation (13) assumes a uniform distribution over conflators. Given these assumptions, the static state distributions π and A can be estimated as:

$$\pi(q) :\approx p\left(W_1 = \tilde{w}_q\right)/n_R \quad (14)$$

$$A(q_1, \ldots, q_k) :\approx p\left(W_i = \tilde{w}_{q_k} | W_{i-k+1}^{i-1} = \tilde{w}_{q_1} \ldots \tilde{w}_{q_{k-1}}\right)/n_R \quad (15)$$

Equations (14) and (15) are nothing more or less than a word k-gram model over extant forms, scaled by the constant $\frac{1}{n_R}$. One 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, a word trigram model ($k = 3$) was trained on the TIGER corpus of contemporary German (Brants et al., 2002). Probabilities for the "unknown" form **u** were computed using the simple smoothing technique of assigning **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, simple maximum likelihood techniques cannot be used to estimate the model's lexical probabilities B_S. Instead, lexical probabilities are instantiated as a Maxwell-Boltzmann distribution for a set d_r of conflator-specific distance functions (Jaynes, 1983):

$$B\big(\langle\tilde{w}, r\rangle, w\big) :\approx \frac{b^{\beta d_r(w,\tilde{w})}}{\sum_{r'\in R}\sum_{\tilde{w}'\in\downarrow[w]_{r'}} b^{\beta d_{r'}(w,\tilde{w}')}} \tag{16}$$

Here, $b, \beta \in \mathbb{R}$ are free model parameters with $b \geq 1$ and $\beta \leq 0$. 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 $\tilde{w}$, and the set $\downarrow[w]_r \subseteq [w]_r \subseteq \mathcal{A}^*$ is a finite set of canonicalization hypotheses provided by r for w, as described in section 2.

It should be explicitly noted that the denominator of the right-hand side of Equation (16) 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 efficiently computed) would adhere to the traditional form $\big(\mathrm{sim}(o,q)/\sum_{o'}\mathrm{sim}(o',q)\big)$ 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 (16) is of the form $\big(\mathrm{sim}(o,q)/\sum_{q'}\mathrm{sim}(o,q')\big)$, which corresponds more closely to a distribution $p(Q|O)$ over *states* conditioned on observations.[8]

From a practical standpoint, it should be clear that Equation (16) is much more efficient to compute than an estimator summing globally over potential observations, since all the data needed to compute Equation (16) 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 admissible historical forms, and hence would be of little practical use. From a formal standpoint, I believe that Equation (16) 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 have identical and uniform density with respect to the d_r, but a proof of this conjecture is beyond the scope of this paper.

It was noted above in Section 2.3 that 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

[8] 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.

computational tractability of Equation (16) therefore, the phonetic conflation hypotheses considered were implicitly restricted to the finite set $\mathcal{W}$ of known extant forms used to define the model's states, $\downarrow[w]_{\text{pho}} = \downarrow_{\mathcal{W}}[w]_{\text{pho}} = [w]_{\text{pho}} \cap \mathcal{W}$. Transliterations and rewrite targets which were not also known extant forms were implicitly mapped to the designated symbol **u** for purposes of estimating transition probabilities for previously unseen extant word types.

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

$$\begin{aligned}
b &= 2 \\
\beta &= -1 \\
R &= \{\text{xlit}, \text{pho}, \text{rw}\} \\
d_{\text{xlit}}(w, \tilde{w}) &= 2/|w| && \text{if } \tilde{w} = \text{xlit}^*(w) \\
d_{\text{pho}}(w, \tilde{w}) &= 1/|w| && \text{if } \tilde{w} \in \downarrow[w]_{\text{pho}} \\
d_{\text{rw}}(w, \tilde{w}) &= [\![M_{\text{rw}} \circ A_{\text{Lex}}]\!](w, \tilde{w})/|w| && \text{if } \tilde{w} = \text{best}_{\text{rw}}(w)
\end{aligned}$$

In all other cases, $d_r(w, \tilde{w})$ is undefined and $B(\langle \tilde{w}, r \rangle, w) = 0$. Note that all conflator distance functions are scaled by inverse input word length $\frac{1}{|w|}$, thus expressing an average distance per input character as opposed to an absolute distance for the input word. 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.[9] The transliteration and phonetic conflators are constants given input word length, whereas the rewrite conflator makes use of the cost $[\![M_{\text{rw}} \circ A_{\text{Lex}}]\!](w, \tilde{w})$ assigned to the conflation pair by the rewrite cascade itself.

3.4 Runtime Disambiguation

Having defined the disambiguator model D, it can be used it to determine a unique "best" canonical form for each input sentence S by application of the well-known *Viterbi algorithm* (Viterbi, 1967). Formally, the Viterbi algorithm computes the state path with maximal probability for the observed sentence:

$$\textsc{Viterbi}(S, D) = \underset{\langle q_1, \ldots, q_{n_S} \rangle \in \mathcal{Q}^{n_S}}{\arg\max} p(q_1, \ldots, q_{n_S}, S|D) \tag{17}$$

Extracting the disambiguated canonical forms $\hat{S} = \langle \hat{w}_1, \ldots, \hat{w}_{n_S} \rangle \in (\mathcal{A}^*)^{n_S}$ from the state sequence $\hat{Q} = \langle \hat{q}_1, \ldots, \hat{q}_{n_S} \rangle = \textsc{Viterbi}(S, D)$ returned by the Viterbi algorithm is a simple matter of projecting the extant word components of the HMM state structures, taking care to map the designated symbol **u** onto an appropriate output

[9] Despite this tendency of natural languages, the combinatorial properties of concatenative monoids dictate that the number of potential "false friends" grows exponentially with input string length if for example arbitrary substitutions are allowed, suggesting an increased likelihood of false positive conflations for *longer* input words. In this context, note that the use of per-character distances results in higher-entropy probability distributions (Shannon, 1948) for longer input strings, effectively treating the d_r distance estimates as increasingly unreliable as input string length grows.

string. Let witness : $\wp(\mathcal{A}^*) \rightarrow \mathcal{A}^*$ be a choice function over conflation hypotheses,[10] witness$(\downarrow[w]_r) \in \downarrow[w]_r$ for all $w \in \mathcal{A}^*, r \in R$ with $\downarrow[w]_r \neq \emptyset$, and for $1 \leq i \leq n_S$, define:

$$\hat{w}_i := \begin{cases} \text{witness}\left(\downarrow[w]_{r_{\hat{q}_i}}\right) & \text{if } \tilde{w}_{\hat{q}_i} = \mathbf{u} \\ \tilde{w}_{\hat{q}_i} & \text{otherwise} \end{cases} \tag{18}$$

Following the equivalence class notation for type-wise conflators, I write $[w_i]_{\text{hmm},D}$ to denote the singleton set $\{\hat{w}_i\}$ containing the unique canonical form returned by the HMM disambiguator D for an input token w_i in sentential context S, omitting the model subscript D where no ambiguity will result.

3.5 Expressive Power

It was noted in section 2 above that each of the type-wise conflators used in the current approach have representations as (weighted) finite-state transducers (WFSTs). Since the union of WFSTs is itself a WFST, as is the concatenation of WFSTs (Mohri, 2009), the type-wise analysis stage which generates canonicalization hypotheses for the disambiguator can be expressed by an extended rational algebraic expression, assuming specialized functions such as the k-best lookup used by the rewrite transducer are included in the inventory of admissible operations. Hidden Markov Models have been shown to be equivalent to the sub-family of WFSTs called probabilistic finite-state automata (PFSAs) by Vidal et al. (2005). Pereira and Riley (1997) advocate a decomposition of HMM component distributions into dedicated WFSTs which may then be cascaded (composed) to simulate the original HMM for use in speech processing applications. Hanneforth and Würzner (2009) present a technique for creating n-gram language models using only the algebra of weighted rational languages which can in principle be extended to implement the disambiguator's dynamic lexical probability distribution given by Equation (16) as just such a dedicated WFST component. Finally, since the Viterbi algorithm can be applied directly to PFSAs (Vidal et al., 2005) and with minimal adaptation to appropriately weighted WFSTs (Mohri, 2002; Jurish, 2010b), the entire proposed canonicalization architecture does not exceed the expressive power of the weighted rational relations.

4 Evaluation

4.1 Test Corpus

The conflation and disambiguation techniques described above were tested on a manually annotated corpus of historical German drawn from the *Deutsches Textarchiv*.[11] 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,

[10] Since conflation hypothesis sets $\downarrow[w]_r$ are finite, the axiom of choice is not strictly required here.
[11] `http://www.deutschestextarchiv.de`

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 human annotator, 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 an information retrieval task Formally, let $C = \{c_1, \ldots, c_{n_C}\}$ be a finite set of canonicalizers, and let $G = \langle g_1, \ldots, g_{n_G} \rangle$ represent the test corpus, where each token g_i is a $(2 + n_C)$-tuple $g_i = \langle w_i, \tilde{w}_i, [w_i]_{c_1}, \ldots, [w_i]_{c_{n_C}} \rangle \in \mathcal{A}^* \times \mathcal{A}^* \times \wp(\mathcal{A}^*)^{n_C}$, for $1 \leq i \leq n_G$. Here, w_i represents the literal token text as appearing in the historical corpus, $\tilde{w}_i$ is its gold-standard canonical cognate, and $[w_i]_{c_j}$ is the set of canonical forms assigned to the token by the canonicalizer c_j, for $1 \leq j \leq n_C$. Let $Q = \bigcup_{i=1}^{n_G} \{\tilde{w}_i\}$ 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 $\text{relevant}(q), \text{retrieved}_c(q) \subset \mathbb{N}$ of *relevant* and *retrieved* corpus tokens as:

$$\text{relevant}(q) = \{i \in \mathbb{N} : q = \tilde{w}_i\} \tag{19}$$

$$\text{retrieved}_c(q) = \{i \in \mathbb{N} : q \in [w_i]_c\} \tag{20}$$

Token-wise precision ($\text{pr}_{\text{tok},c}$) and recall ($\text{rc}_{\text{tok},c}$) for the canonicalizer c can then be defined as:

$$\text{pr}_{\text{tok},c} = \frac{\left|\bigcup_{q \in Q} \text{retrieved}_c(q) \cap \text{relevant}(q)\right|}{\left|\bigcup_{q \in Q} \text{retrieved}_c(q)\right|} \tag{21}$$

$$\text{rc}_{\text{tok},c} = \frac{\left|\bigcup_{q \in Q} \text{retrieved}_c(q) \cap \text{relevant}(q)\right|}{\left|\bigcup_{q \in Q} \text{relevant}(q)\right|} \tag{22}$$

Type-wise measures $\text{pr}_{\text{typ},c}$ and $\text{rc}_{\text{typ},c}$ are defined analogously, by mapping the token index sets of Equations (19) and (20) to corpus types before applying Equations (21) and (22). I use the unweighted harmonic precision-recall average F (van Rijsbergen,

	% 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 with respect to the *Deutsches Textarchiv* evaluation subset. The maximum value in each column appears in boldface type.

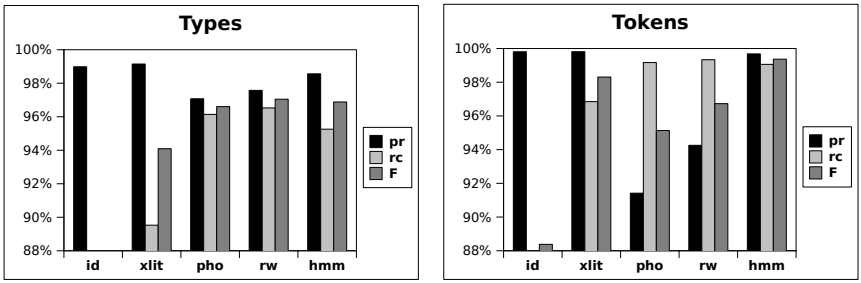

Figure 2: Evaluation data for various canonicalization techniques: visualization

1979) as a composite measure for both type- and token-wise evaluation modes:

$$\mathrm{F}(pr, rc) \quad = \quad \frac{2 \cdot pr \cdot rc}{pr + rc} \tag{23}$$

I follow Charniak et al. (1993) in using *relative error reduction rates* rather than absolute differences when comparing the performance of different canonicalizers. The general form for the (relative) error reduction in evaluation mode x provided by a method c_2 over method c_1 is: $\frac{x_{c_2} - x_{c_1}}{1 - x_{c_1}}$, assuming $0 \leq x_{c_1} \leq x_{c_2} \leq 1$. For example, given the data in Table 1, the error reduction in type-wise recall $x = \mathrm{rc}_{\mathrm{typ}}$ provided by $c_2 = \mathrm{rw}$ with respect to $c_1 = \mathrm{xlit}$ is $\frac{\mathrm{rc}_{\mathrm{typ,rw}} - \mathrm{rc}_{\mathrm{typ,xlit}}}{1 - \mathrm{rc}_{\mathrm{typ,xlit}}} = \frac{.965 - .895}{1 - .895} \approx 0.67 = 67\%$.

4.3 Results

Evaluation results for the canonicalization techniques described in sections 2 and 3 with respect to the test corpus are given in Table 1 and graphically depicted in Figure 2. 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% of type-wise precision errors and of over 94% of 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 type-wise harmonic average F falls by a mere 0.1% in absolute magnitude (3% relative error introduction) with respect to the highest-recall method (rw). Indeed, the token-wise composite measure F is substantially higher for the HMM disambiguator (99.4%, vs. 96.7% for the rewrite method), with an error reduction rate of over 64% compared to its closest competitor, deterministic transliteration (xlit).

The most surprising aspect of these results is the recall performance of the conservative transliterator xlit with $\mathrm{rc}_{\mathrm{tok}} = 96.8\%$, reducing token-wise recall errors by over 84% compared to the naïve string identity method. 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, whereas diachronically more heterogeneous corpora have been shown to be less amenable to such simple techniques (Gotscharek et al., 2009b; Jurish, 2010a).

5 Conclusion

I 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 fail to associate many historical spelling variants with any appropriate contemporary cognate at all. 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, I proposed a method for disambiguating canonicalization hypotheses at the token level using sentential context to optimize the path probability of candidate canonical forms given the observed historical forms. The disambiguator uses a Hidden Markov Model whose lexical probabilities are dynamically re-computed for every input sentence based on the canonicalization hypotheses returned by a set of subordinated type-wise conflators, the entire canonicalization cascade remaining within the domain of weighted rational transductions.

The proposed disambiguation architecture was evaluated on an information retrieval task over a gold standard corpus of manually confirmed canonicalizations of historical

German text drawn from the *Deutsches Textarchiv*. Use of the token-wise disambiguator provided a relative precision error reduction of over 94% with respect to the best recall method, and a relative 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 harmonic precision-recall average $\mathrm{F} = 99.4\%$.

I am interested in verifying these 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 particular, I am interested in comparing the performance of the manually constructed rewrite transducer used here with a linguistically motivated language-independent conflator (Covington, 1996; Kondrak, 2000) on the one hand, and with conflators induced from a training sample by machine learning techniques (Ristad and Yianilos, 1998; Kempken et al., 2006; Ernst-Gerlach and Fuhr, 2006) on the other. Future work on the disambiguator itself should involve a systematic investigation of the effects of the various model parameters as well as more sophisticated smoothing techniques for handling previously unseen extant types and sparse training data.

Acknowledgements

The work described here was funded by a *Deutsche Forschungsgemeinschaft* (DFG) grant to the project *Deutsches Textarchiv*. Additionally, the author would like to thank Henriette Ast, Jörg Didakowski, Marko Drotschmann, Alexander Geyken, Susanne Haaf, Thomas Hanneforth, Wolfgang Seeker, Kay-Michael Würzner, and this paper's anonymous reviewers for their helpful feedback and comments.

References

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

Black, A. W. and Taylor, P. (1997). Festival speech synthesis system. Technical Report HCRC/TR-83, University of Edinburgh, Centre for Speech Technology Research.

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

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

Brill, E. (1992). A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92*, pages 152–155.

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

Cafarella, M. and Cutting, D. (2004). Building Nutch: Open source search. *Queue*, 2(2):54–61.

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

Church, K. W. and Gale, W. A. (1991). Probability scoring for spelling correction. *Statistics and Computing*, 1:93–103.

Covington, M. A. (1996). An algorithm to align words for historical comparison. *Computational Linguistics*, 22:481–496.

Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7:171–176.

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

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

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

Gadd, T. N. (1988). 'Fisching fore werds': phonetic retrieval of written text in information systems. *Program*, 22(3):222–237.

Gadd, T. N. (1990). PHONIX: The algorithm. *Program*, 24(4):363–366.

Geyken, A. and Hanneforth, T. (2006). TAGH: A complete morphology for German based on weighted finite state automata. In *Proceedings FSMNLP 2005*, pages 55–66, Berlin. Springer.

Gotscharek, A., Neumann, A., Reffle, U., Ringlstetter, C., and Schulz, K. U. (2009a). 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. ACM.

Gotscharek, A., Reffle, U., Ringlstetter, C., and Schulz, K. U. (2009b). On lexical resources for digitization of historical documents. In *Proceedings of the 9th ACM symposium on Document Engineering*, DocEng '09, pages 193–200, New York. ACM.

Hanneforth, T. and Würzner, K.-M. (2009). Statistical language models within the algebra of weighted rational languages. *Acta Cybernetica*, 19(2):313–356.

Jaynes, E. T. (1983). Brandeis lectures. In *E. T. Jaynes: Papers on Probability, Statistics and Statistical Physics*, pages 40–76. D. Reidel, Dordrecht.

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

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

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

Jurish, B. (2010a). Comparing canonicalizations of historical German text. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology* (SIGMORPHON), pages 72–77.

Jurish, B. (2010b). Efficient online k-best lookup in weighted finite-state cascades. In Hanneforth, T. and Fanselow, G., editors, *Language and Logos: Studies in Theoretical and Computational Linguistics*, volume 72 of *Studia grammatica*, pages 313–327. Akademie Verlag, Berlin.

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

Kempken, S. (2005). *Bewertung von historischen und regionalen Schreibvarianten mit Hilfe von Abstandsmaßen*. Diploma thesis, Universität Duisburg-Essen.

Kempken, S., Luther, W., and Pilz, T. (2006). Comparison of distance measures for historical spelling variants. In Bramer, M., editor, *Artificial Intelligence in Theory and Practice*, pages 295–304. Springer, Boston.

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

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

Kondrak, G. (2002). *Algorithms for Language Reconstruction*. PhD thesis, University of Toronto.

Kukich, K. (1992). Techniques for automatically correcting words in texts. *ACM Computing Surveys*, 24(4):377–439.

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

Lidstone, G. J. (1920). 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.

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

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Mays, E., Damerau, F. J., and Mercer, R. L. (1991). Context based spelling correction. *Information Processing & Management*, 27(5):517–522.

Möhler, G., Schweitzer, A., and Breitenbücher, M. (2001). *IMS German Festival manual, version 1.2*. Institute for Natural Language Processing, University of Stuttgart.

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

Mohri, M. (2009). Weighted automata algorithms. In *Handbook of Weighted Automata*, Monographs in Theoretical Computer Science, pages 213–254. Springer, Berlin.

Pereira, F. C. N. and Riley, M. D. (1997). Speech recognition by composition of weighted finite automata. In Roche, E. and Schabes, Y., editors, *Finite-State Language Processing*, pages 431–453. MIT Press, Cambridge, MA.

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

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

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

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

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

Rayson, P., Archer, D., and Smith, N. (2005). 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.

Ristad, E. S. and Yianilos, P. N. (1998). Learning string edit distance. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 20(5):522–532.

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

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

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

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423.

Sokirko, A. (2003). A technical overview of DWDS/dialing concordance. Talk delivered at the meeting *Computational linguistics and intellectual technologies*, Protvino, Russia.

Unicode Consortium (2011). *The Unicode Standard*. The Unicode Consortium, Mountain View, CA.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA.

Verberne, S. (2002). *Context-sensitive spell checking based on word trigram probabilities*. Master thesis, University of Nijmegen.

Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., and Carrasco, R. C. (2005). Probabilistic finite-state machines – Part II. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1026–1039.

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

Zielinski, A., Simon, C., and Wittl, T. (2009). Morphisto: Service-oriented open source morphology for German. In Mahlow, C. and Piotrowski, M., editors, *State of the Art in Computational Morphology*, pages 64–75. Springer, Berlin.