Dsolve – Morphological Segmentation for German using Conditional Random Fields*

Kay-Michael Würzner and Bryan Jurish
Berlin-Brandenburg Academy of Sciences and Humanities

— DRAFT —

Abstract

We describe Dsolve, a system for the segmentation of morphologically complex German words into their constituent morphs. Our approach treats morphological segmentation as a classification task, in which the locations and types of morph boundaries are predicted by a Conditional Random Field model trained from manually annotated data. The prediction of morph-boundary types in addition to their locations distinguishes Dsolve from similar approaches previously suggested in the literature. We show that the use of boundary types provides a (somewhat counter-intuitive) performance boost with respect to the simpler task of predicting only segment locations.

1 Introduction

The goal of the morphological segmentation of words is their decomposition into *morphemes* (lexical level) or *morphs* (text level), each of which may be associated with a lexical meaning and/or a grammatical function. The segmentation of a word into morphemes is often referred to as *deep* segmentation, and is contrasted to *surface-level* segmentation into morphs (cf. e.g. [5]). Given the segmentation of the German compound *Ärztekammern* (engl. "medical associations") into *Ärzt-e-Kammer-n*, the difference between the deep and the surface levels is observable in the segment *Ärzt*, which is a variant of the noun *Arzt* (engl. "doctor") which may only be realized in the plural. *Arzt* and *Ärzt* are distinct surface realizations – called *allomorphs* – of the morpheme {Arzt}. The task of surface-level morphological segmentation of a word can be viewed as identification on the one hand of the word formation operations which contribute to the word's construction (i.e. *compounding*, *derivation*, *inflection*) and of the morphs which constitute the operands of these operations on the other.

Morphological segmentations have many applications in (computational) linguistics, including information retrieval [19], language learning [2], and letter-to-sound conversion [6]. In the following, we present Dsolve, a system for surface-level segmentation of words based on supervised training of a conditional random field model (CRF) [17]. In order to classify our approach, we first give an overview of the related literature. We then describe Dsolve in more detail, evaluate its performance on a modest set of manually annotated German words, and conclude with some insights and loose ends.

*Originally appeared in C. Mahlow and M. Piotrowski (Eds.), *Systems and Frameworks for Computational Morphology (SFCM 2015)*, CCIS 537, pp. 94–103, 2015. The final publication is available at Springer via `http://dx.doi.org/10.1007/978-3-319-23980-4_6`.

2 Related Work

Approaches to automatic morphological segmentation can be classified based on two characteristics. The first of these is the distinction between methods which rely on the manual specification of potential morph(eme) combinations – typically using some grammar formalism — and those which automatically induce such knowledge from un-annotated language data using distributional inference. The second characteristic is whether the approach in question makes use of some list or lexicon of (free) morphemes or not. Such a list can substantially increase quality of the results, but is often associated with a great deal of manual effort.

Traditional finite-state or two-level approaches [1] use manually constructed rules together with an extensive lexicon. The initial cost of creating a reliable system is thus quite high. The scope of finite-state morphologies usually includes other applications such as base-form reduction, word categorization, or lexical-semantic analysis. This broad scope often implies the inclusion of morphologically complex forms into the underlying lexica, which adversely affects their performance on the task of morphological segmentation. For German, the most prominent systems are GERTWOL [12], SMOR [27], and TAGH [9], which are either closed-source commercial products (GERTWOL and TAGH) or suffer from the aforementioned over-lexicalization issue (SMOR and TAGH).

A great number of proposals have been made to reduce the effort involved in the creation of morphological analysis systems. One class of proposals makes use of the fact that the number of affixes is much smaller than the number of free morphemes. Such *affix removal stemmers* [8] successively remove known affixes from words under the assumption that the remaining string is the word stem [22, 24].

In statistical methods, linguistic knowledge is represented as a probability distribution over some atomic unit which is inferred from language data. In unsupervised settings (sometimes called *morphology induction*), these data are raw (i.e. unannotated) texts. Important works in this category include [7, 10, 3, 4]. All of these approaches have in common that they aim at constructing lists of possible morph(eme)s using a set of pre-defined heuristics. The identification of morph(eme)s occurs by means of reference to character or string frequencies. In [10] for example, an underlying word structure $prefix?\ stem\ suffix*$ is assumed in order to identify so-called *signatures*, classes of stems which occur with common affixes. The performance of systems trained in an unsupervised manner is surprisingly high (compare for example the results of *Morpho Challenge*, [16]), but still insufficient for productive applications.

The inclusion of manually segmented words in the training process (so called semi-supervised settings) can improve the performance of statistical methods dramatically [15]. An early instance of a data-driven approach which is trained in a completely supervised manner is MOSES [14]. For each bigram of adjacent characters in a word, the most likely intervening boundary type is selected on the basis of the boundary type distributions in the training material. Following this idea, there have been a number of proposals which focus on modeling the relation between local substrings and morph(eme) boundaries. Dsolve itself falls into this category, and we elaborate on the precise nature of the relation to be modeled below.

CRFs have previously been applied to word segmentation in Chinese in [29]. Studies on morphological segmentation of languages with alphabetic writing systems were carried out on Arabic [11, 25], English [2, 25, 26], Finnish [25, 26], Hebrew [25] and Turkish [25, 26].

	G	e	f	o	l	g	s	l	e	u	t	e	n
(a)	0	1	0	0	0	1	1	0	0	0	0	1	0
(b)	0	+	0	0	0	˜	#	0	0	0	0	˜	0

Figure 1: The German noun compound *Gefolgsleuten* (engl. "henchmen$_{\text{[DATIVE]}}$") boundary classified using (a) a binary classification scheme and (b) a type-sensitive classification scheme.

3 Morphological Segmentation as Sequence Classification

Sequence classification is a popular technique in natural language processing, already having been used successfully e.g. for tokenization, part-of-speech tagging, and named entity recognition. At its core, the sequence classification task is defined in terms of a given set of symbols O and a set of classes C, and maps each symbol $o_i \in O$ in an observation string $o = o_1 \dots o_n$ onto a class $c_i \in C$ by determining the most probable string of classes $c = c_1 \dots c_n$ associated with o by an underlying stochastic model. Individual statistical models differ in the manner in which the most probable classes are calculated. Hidden Markov models for example optimize the joint probability $P(o, c)$ [23], while CRFs optimize the conditional probability $P(c|o)$ [31].[1]

For the task of morphological segmentation, the set of symbols is simply the surface character alphabet itself (or the set of character N-grams over this alphabet [14]). The set of target classes is usually two-valued (e.g. $C = \{0, 1\}$), leading to a classifier which predicts for every position i whether or not there is a segment boundary following (rsp. preceding) the observed symbol at position i of the input word, as illustrated in Figure 1a.

Some approaches (e.g. [14, 25]) use more complex classification schemes in order to define morph(eme)s as *spans* in words (e.g. $C = \{B, I, E, S\}$ with B indicating the initial character of a multi-character morph, I a character inside a multi-character morph, E the final character of a multi-character morph, and S a single-character morph. Since such a span-based classification may result in inconsistent predictions, further disambiguation heuristics are required to interpret the classification results.

The mere detection of boundaries is itself however insufficient for some applications. Consider for example the task of syllabification which in most languages follows the maximum onset principle [28]. In German, the morphological structure of words overrides this principle in cases of prefixion and compounding. Dsolve attempts to accommodate such phenomena by using a type-sensitive classification scheme: $C_{\text{Dsolve}} = \{+, \#, \sim, 0\}$, where '+' indicates that a prefix morph ends at the current position, '#' indicates that a free morph starts with the following position, '$\sim$' indicates that a suffix morph starts with the following position, and 0 indicates that there is no morph boundary after the current position. An example using this classification scheme is given in Figure 1b.

We chose CRFs as the computational framework for the classification task. CRFs are a class of stochastic models using chain-structured undirected graphs to encode the dependencies between observations and output labels (i.e. classes). These dependencies are expressed in terms of feature functions representing salient properties of the input. Feature functions

[1] Although as correctly noted in [23], any class-string c which maximizes $P(c, o)$ will also maximize $P(c|o)$ if the observation string o is held fixed.

depending on external data sources, distributional properties such as "successor frequency" [13] as extracted from a large corpus of (un-annotated) data [26], or the distinction between vowels and consonants [2] have also been proposed for the current task.

In the case of Dsolve, we defined a simple feature inventory using only unigram features based on local string context. Each position i in the input string $o = o_1 \dots o_n$ is assigned a feature for each substring of o of length $m \leq N$ within a context window of $N-1$ characters relative to position i (including implicit word boundary symbols with pseudo-indices 0 and $n+1$). Formally, a Dsolve model of "order"[2] N has $2N^2 - \sum_{m=1}^{N} m$ distinct feature functions f_j^k, where $-N < j \leq k < N$ and $k - j < N$, with $f_j^k(o_i) = o_{i+j} o_{i+j+1} \cdots o_{i+k-1} o_{i+k}$. A model with $N = 3$ for example has 12 distinct feature functions. If $o = sport$, then o_3 has the non-zero features $\{f_{-2}^{-2} = `s\text{'}, f_{-2}^{-1} = `sp\text{'}, f_{-2}^{0} = `spo\text{'}, f_{-1}^{-1} = `p\text{'}, f_{-1}^{0} = `po\text{'}, f_{-1}^{1} = `por\text{'}, f_0^0 = `o\text{'}, f_0^1 = `or\text{'}, f_0^2 = `ort\text{'}, f_1^1 = `r\text{'}, f_1^2 = `rt\text{'}, f_2^2 = `t\text{'}\}$. During model training, the influence of each feature expressed as a real-valued weight is optimized with respect to a manually classified training set. For the current experiments, optimization was performed by means of the L-BGFS algorithm [20].

4 Evaluation

In this section, we investigate the influence of model order on the tasks of boundary detection and optional classification of word-internal morph boundaries. We report model performance in terms of *string accuracy* (acc), *precision* (pr), *recall* (rc), and the and unweighted precision-recall harmonic average F [30]. For evaluation purposes, given a finite set W of annotated words and a finite set C of boundary classes with $0 \in C$ the designated non-boundary class, we associate with each word $w = w_1 w_2 \dots w_m \in W$ a partial *relevant* boundary-placement function $B_{\text{relevant},w} : \mathbb{N} \to C \backslash \{0\}$ such that $B_{\text{relevant},w}(i) = c$ if and only if there exists a manually annotated morph boundary of type $c \in C$ in the word w between the characters w_{i-1} and w_i, $1 < i \leq m$. The *retrieved* morph-boundary placement function $B_{\text{retrieved},w}$ is defined analogously based on the output of the CRF labelling. The evaluation quantities for the detection and classification task can then be defined in the usual manner:

$$\text{pr} = |\text{relevant} \cap \text{retrieved}| \,/\, |\text{retrieved}| \tag{1}$$

$$\text{rc} = |\text{relevant} \cap \text{retrieved}| \,/\, |\text{relevant}| \tag{2}$$

$$\text{F} = (2 \cdot \text{pr} \cdot \text{rc}) \,/\, (\text{pr} + \text{rc}) \tag{3}$$

$$\text{acc} = |\{w \in W \mid B_{\text{retrieved},w} = B_{\text{relevant},w}\}| \,/\, |W| \tag{4}$$

where:

$$\text{relevant} = \bigcup_{w \in W} \{w\} \times B_{\text{relevant},w} = \{(w,i,c) \mid (i \mapsto c) \in B_{\text{relevant},w}\} \tag{5}$$

$$\text{retrieved} = \bigcup_{w \in W} \{w\} \times B_{\text{retrieved},w} = \{(w,i,c) \mid (i \mapsto c) \in B_{\text{retrieved},w}\} \tag{6}$$

Evaluators for the detection-only task can be defined identically, after mapping all boundary types $c \in C \backslash \{0\}$ to a single, shared value.

Boundary Symbol	pr%	rc%	F%	acc%
+	92.05	97.20	94.56	n/a
#	96.01	93.28	94.63	n/a
$\sim$	93.28	92.66	92.97	n/a
TOTAL[+types]	93.74	93.74	93.74	87.40
TOTAL[−types]	96.20	96.20	96.20	87.40

Table 1: Comparison of two independent manual segmentations of a sample of Dsolve's training materials.

4.1 Materials

We created a list of 15,522 distinct German word-forms and manually annotated types and locations of all word-internal morph boundaries. In unclear cases, we consulted `canoo.net` and/or the *Etymologisches Wörterbuch des Deutschen* [21]. If multiple correct segmentations were applicable, we randomly selected one of them. Candidate word-forms were selected from various corpora in the collection of the *Zentrum Sprache* at the Berlin-Brandenburg Academy of Sciences and Humanities[3]. A total of 21,068 word-internal morph boundaries were annotated in this fashion. In the interest of providing an accurate approximation of the morph boundary distribution in German and to guard against false-positive boundary predictions, 3,555 monomorphemic words were also included in the list. The complete list is published under the terms of the CC BY-SA 3.0 license, and is available for download at `http://kaskade.dwds.de/$\sim$moocow/gramophone/de-dlexdb.data.txt`. In order to provide an assessment of the plausibility of our segmentations, we created explicit written annotation guidelines and asked a professional linguist of our acquaintance to edit a sample of 1,000 words accordingly. The results of a comparison to our segmentation is shown in Table 1, where our segmentations as above are interpreted as "relevant" boundaries, and the independent third-party segmentations provide the "retrieved" boundaries. Training and run-time application of CRFs were performed with the `wapiti` toolkit [18].

4.2 Method

The evaluation data was randomly partitioned into ten chunks of approximately equal size and evaluated by 10-fold cross-validation. For each of the ten training subsets and for each model order N with $1 \leq N \leq 5$, we trained two CRF model variants using a context window of N characters for CRF model features as described in Section 3. The first model variant, which we designate with the subscript [+types] predicts both boundary location and type by internal use of 3 distinct boundary labels for prefix-, stem-, and suffix-boundaries, respectively, in addition to a designated label for non-boundaries. The second model variant, indicated by the subscript [−types], uses only two labels indicating the presence or absence of a morph boundary, regardless of its type. For purposes of comparison, we also included results on the

[2]Note that our use of "model order" in this paper refers only to the context window size used to define the feature function inventory, and is unrelated to the order of linear-chain feature dependencies in the underlying CRF models.

[3]`http://www.bbaw.de`

N	pr%	rc%	F%	acc%
1	27.27	0.01	0.03	22.84
2	70.84	60.92	65.51	47.29
3	85.23	82.64	83.91	70.60
4	91.39	88.77	90.07	80.50
5	93.46	90.67	92.04	83.50

Table 2: Evaluation results for Dsolve on the combined boundary-detection and classification task for [+types] model variants.

	Prefix-Stem (+)			**Stem-Stem** (#)			**Stem-Suffix** (∼)		
N	**pr%**	**rc%**	**F%**	**pr%**	**rc%**	**F%**	**pr%**	**rc%**	**F%**
1	–	0.00	–	27.27	0.05	0.10	–	0.00	–
2	63.97	50.25	56.28	71.47	51.27	59.71	72.65	69.83	71.21
3	83.62	85.65	84.63	87.27	77.31	81.99	84.89	84.31	84.60
4	92.44	92.35	92.39	93.04	86.07	89.42	90.21	88.87	89.54
5	95.57	94.68	95.12	95.01	88.83	91.81	91.92	90.16	91.03

Table 3: Detailed results for Dsolve boundary classification by boundary type.

boundary detection task for both Morfessor FlatCat[4] as well as a `wapiti` re-implementation of the span-based CRF model described in [25] as methods "FlatCat" and "spanCRF", respectively. Each trained model was applied to the respective disjoint test subset, and the evaluation quantities defined above were computed for the concatenation of all test subsets.

4.3 Results & Discussion

Evaluation results for the joint task of boundary detection and classification are given in Tables 2 and 3, and results for the boundary detection task modulo classification are given in Table 4. Note that since the [−types] model variants were incapable of predicting boundary classes, they were not considered for the joint detection and classification task.

The most prominent effect observable in the data is the fact that all evaluation quantities increase monotonically as model order grows. Such a tendency is common for n-gram models of natural language phenomena, and can be interpreted in the current case as a lexicalization effect: as model order grows, the induced models are able to incorporate information on the distributions of increasingly long whole morphs. This hypothesis is supported on the one hand by the data from Table 3, indicating that the induced models performed most poorly for strong (#) morph boundaries – which necessarily occur between comparatively long stem morphs – and on the other hand by the disproportionate performance gain of the models with orders 2 and 3 with respect to their predecessors, since most German prefixes and suffixes are of length 2 or 3.

Unsurprisingly, comparing the quantities for the [+types] model variants from Tables 2 and 4 shows that the task of morph boundary detection is in some sense easier than the joint

[4] `http://www.cis.hut.fi/projects/morpho/morfessorflatcat.shtml`; FlatCat models were trained with perplexity threshold 10.0 using annotated corpus data in semi-supervised mode.

Method	Variant	N	pr%	rc%	F%	acc%
FlatCat	–	–	79.18	89.48	84.01	75.27
spanCRF	–	1	40.33	9.57	15.47	24.13
spanCRF	–	2	77.35	71.80	74.47	55.04
spanCRF	–	3	88.43	87.52	87.97	74.49
spanCRF	–	4	92.83	91.33	92.08	82.57
spanCRF	–	5	93.56	92.29	92.92	84.45
Dsolve	+types	1	36.36	0.02	0.04	22.84
Dsolve	+types	2	79.45	68.32	73.47	53.16
Dsolve	+types	3	89.36	86.64	87.98	74.35
Dsolve	+types	4	93.49	90.81	92.13	82.55
Dsolve	+types	5	94.46	91.63	93.02	84.36
Dsolve	−types	1	56.34	0.72	1.42	23.03
Dsolve	−types	2	77.53	69.61	73.36	52.94
Dsolve	−types	3	88.81	86.58	87.68	73.70
Dsolve	−types	4	92.93	90.78	91.85	81.92
Dsolve	−types	5	93.89	91.73	92.80	83.98

Table 4: Evaluation results on the boundary-detection task for Dsolve variants both with (+types]) and without (−types) model-internal use of distinct boundary classes.

task of boundary detection and classification. Since both the models and data-set partitions used for evaluation were identical, the observed differences are clearly due to the fact that some "errors" in the joint task arose from incorrect predictions of boundary types, albeit at the correct positions: $B_{\text{relevant},w}(i) = c \neq c' = B_{\text{retrieved},w}(i)$.

Both of the Dsolve model variants as well as the spanCRF models substantially outperformed the Morfessor FlatCat baseline for all model orders $N > 3$. The Dsolve[+types] model variants performed quite similarly to the closely related spanCRF models. For $N > 1$, the Dsolve[+types] models were slightly more precise than the spanCRF models of the same order, while the latter achieved slightly higher recall rates. Since the Dsolve[+types] errors were more uniformly distributed between false positives and false negatives for $N > 3$, these achieved a higher harmonic average F than their spanCRF counterparts, although the latter were slightly more successful in terms string accuracy. Due to the limited size of the test corpus, differences on the order of magnitude observed between the Dsolve[+types] and spanCRF models must be viewed with a modicum of skepticism: the differences for $N = 5$ for example stem from a total of only 136 boundary errors and 13 string errors.

Despite the simplicity of the detection-only task, the Dsolve[+types] model variants making use of distinct boundary classes consistently outperformed the [−types] variants using a only binary label set for all model orders $N > 1$ in terms of both precision and string accuracy, leading to a relative error reduction of 9.33% for precision at model order $N = 5$. While the [−types] variants displayed a slightly improved recall in some cases, the effect was not sufficient to outperform the [+types] models on either of the "top-level" evaluation quantities F or string accuracy for nontrivial model orders $N > 1$. This somewhat counter-intuitive effect can only be attributed to the use of multiple boundary classes in the [+types] variants: since the underlying CRF models allow not only the *presence* of a boundary but also its *class* to influence the conditional path probability, these models are capable of capturing distributional regularities beyond those available to the [−types] models, which only encode boundaries' presence or absence. Postulation of a prefix-boundary for example allows a [+types] model to abstract over the lexical content of the prefix in question when estimating subsequent path probabilities, whereas a [−types] model would require additional surface context in order to identify the prefix morph as such and adjust its predictions accordingly.

5 Conclusion & Outlook

We have presented a system for the surface segmentation of morphologically complex words. Treating segmentation as a classification task, our approach uses a Conditional Random Field model trained on a modest set of manually annotated data to predict both the locations and the respective types of morph boundaries for each new input word. Evaluation by cross-validation on a list of 15,522 manually annotated German word-forms showed promising results, with a model using a context window of $N = 5$ input characters achieving a total precision-harmonic average $F \approx 93\%$ on a joint boundary detection and classification task. Somewhat surprisingly, the incorporation of multiple distinct boundary classes into the CRF model was shown to provide a performance gain on a boundary detection task when compared to an otherwise equivalent model encoding only boundary presence or absence. We attribute this effect to the classification models' greater ability to represent linguistically salient distributional regularities.

We are interested in applying our approach to other languages and data-sets, and in extending the approach as presented above by the optional inclusion of user-supplied lexical data (e.g. lists of known prefixes, stems, and/or suffixes). Future work should also investigate to what degree if any the model training phase can be augmented by semi-supervised learning techniques [15] using a large corpus of un-annotated data.

References

[1] K. R. Beesley and L. Karttunen. *Finite State Morphology.* CSLI, Stanford, CA, 2003.

[2] J. Z. Chang and J. S. Chang. Word root finder: a morphological segmentor based on CRF. In *Proceedings of COLING 2012: Demonstration Papers*, pages 51–58, 2012.

[3] M. Creutz and K. Lagus. Unsupervised discovery of morphemes. In *Proceedings of the ACL-2002 Workshop on Morphological and Phonological Learning*, pages 21–30, 2002.

[4] M. Creutz and K. Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1):3:1–3:34, 2007.

[5] M. Creutz and K. Lindén. Morpheme segmentation gold standards for Finnish and English. Technical Report A77, Helsinki University of Technology, 2004.

[6] W. Daelemans. Grafon: a grapheme-to-phoneme conversion system for Dutch. pages 133–138, 1988.

[7] H. Déjean. Morphemes as necessary concept for structures discovery from untagged corpora. In *Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning*, pages 295–298, 1998.

[8] W. B. Frakes. Stemming algorithms. In W. B. Frakes and R. Baeza-Yates, editors, *Information retrieval*, pages 131–160. Prentice-Hall, Upper Saddle River, NJ, 1992.

[9] 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*, pages 55–66. Springer, 2006.

[10] J. Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198, 2001.

[11] S. Green and J. DeNero. A class-based agreement model for generating accurately inflected translations. In *Proceedings of ACL-2012*, pages 146–155, 2012.

[12] M. Haapalainen and M. Ari. GERTWOL und morphologische Disambiguierung für das Deutsche. In *Proceedings of the 10th Nordic Conference of Computational Linguistics*. University of Helsinki, Department of General Linguistics, 1995.

[13] Z. Harris. From phoneme to morpheme. *Language*, 31:190–222, 1955.

[14] U. Klenk and H. Langer. Morphological segmentation without a lexicon. *Literary and Linguistic Computing*, 4(4):247–253, 1989.

[15] O. Kohonen, S. Virpioja, and K. Lagus. Semi-supervised learning of concatenative morphology. In *Proceedings of SIGMORPHON-2010*, pages 78–86, 2010.

[16] M. Kurimo, S. Virpioja, V. Turunen, and K. Lagus. Morpho Challenge Competition 2005–2010: Evaluations and results. In *Proceedings of SIGMORPHON-2010*, pages 87–95, 2010.

[17] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, 2001.

[18] T. Lavergne, O. Cappé, and F. Yvon. Practical very large scale CRFs. In *Proceedings of ACL-2010*, pages 504–513, 2010.

[19] C. Müller and I. Gurevych. Semantically enhanced term frequency. In C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, S. Rüger, and K. van Rijsbergen, editors, *Advances in Information Retrieval*, pages 589–601, Berlin, 2010. Springer.

[20] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, Berlin, 1999.

[21] W. Pfeifer. *Etymologisches Wörterbuch des Deutschen*. Akademie-Verlag, Berlin, 2nd edition, 1993.

[22] M. F. Porter. An algorithm for suffix stripping. *Electronic Library and Information Systems*, 14(3):130–137, 1980.

[23] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[24] U. D. Reichel and K. Weilhammer. Automated morphological segmentation and evaluation. In *Proceedings of LREC*, pages 503–506, 2004.

[25] T. Ruokolainen, O. Kohonen, S. Virpioja, and M. Kurimo. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 29–37, 2013.

[26] T. Ruokolainen, O. Kohonen, S. Virpioja, and M. Kurimo. Painless semi-supervised morphological segmentation using conditional random fields. In *Proceedings of EACL-2014*, pages 84–89, 2014.

[27] H. Schmid, A. Fitschen, and U. Heid. SMOR: A German computational morphology covering derivation, composition and inflection. In *Proceedings of LREC*, 2004.

[28] E. O. Selkirk. On the nature of phonological representation. In T. Myers, J. Laver, and J. Anderson, editors, *The Cognitive Representation of Speech*, pages 379–388. North-Holland Publishing Company, 1981.

[29] H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning. A conditional random field word segmenter for SIGHAN bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, 2005.

[30] C. J. van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, 1979.

[31] H. M. Wallach. Conditional random fields: An introduction. Technical Report MS-CIS-04-21, University of Pennsylvania, Department of Computer and Information Science, 2004.