Morphological Analysis without Expert Annotation

Garrett Nicolai and **Grzegorz Kondrak**
Department of Computing Science
University of Alberta, Edmonton, Canada
{nicolai,gkondrak}@ualberta.ca

Abstract

The task of morphological analysis is to produce a complete list of lemma+tag analyses for a given word-form. We propose a discriminative string transduction approach which exploits plain inflection tables and raw text corpora, thus obviating the need for expert annotation. Experiments on four languages demonstrate that our system has much higher coverage than a hand-engineered FST analyzer, and is more accurate than a state-of-the-art morphological tagger.

1 Introduction

The task of morphological analysis is to annotate a given word-form with its lemma and morphological tag. Since word-forms are often ambiguous, the goal is to produce a complete list of correct analyses, which may involve not only multiple inflections, but also distinct lemmas and parts of speech (c.f. Table 1). Hand-built lexicons, such as CELEX (Baayen et al., 1995), contain this kind of information, but they exist only for a small number of languages, are expensive to create, and have limited coverage. Finite-state analyzers, such as Morphisto (Zielinski and Simon, 2009) and Omorfi (Pirinen, 2015), provide an alternative to lexicons, but their construction also requires expert knowledge and substantial engineering effort. Furthermore, they are often more general than lexicons, although they may require a lemmatic lexicon to ensure high precision.

Morphological tagging is a distinct but related task, which aims at determining a single correct analysis of a word-form within the context of a sentence. Machine learning taggers, such as Morfette (Chrupała et al., 2008) and Marmot (Müller et al., 2013), are capable of achieving high tagging accuracy, but they need to be trained on morphologically annotated corpora, which are unavailable for most languages. Often, morphological tagging can be performed as a downstream application of morphological analysis: tools such as Marmot and the Zurich Dependency Parser (Sennrich et al., 2009) have the functionality to incorporate the output of a morphological analyzer to perform morphological tagging.

In this paper, we propose a novel discriminative string transduction approach to morphological analysis, which is designed to be trained on plain inflection tables, thus obviating the need for expert rule engineering or morphologically annotated corpora. Inflection tables are available for many languages on web sites such as Wiktionary, thanks to crowd-sourcing efforts of moderately-skilled native speakers.[1] In addition, our system is capable of leveraging raw unannotated corpora to refine its analyses by re-ranking. The accuracy of the system on German approaches that of a hand-engineered FST analyzer, while having much higher coverage. The experimental results on English, Dutch, German, and Spanish demonstrate that it also more accurate than the analysis module of a state-of-the-art morphological tagger.

2 Methods

Our approach to morphological analysis is based on string transduction between a word-form (e.g. *lüfte*) and an analysis composed of a lemma and a tag (e.g. *lüften*+1SIE), where the tag corresponds to the predicted inflection slot. Our system consists of four modules: alignment, transduction, reranking, and thresholding.

[1] The Unimorph Project (*unimorph.org*) provides inflection tables for more than 350 languages.

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 211–216,
Valencia, Spain, April 3-7, 2017. ©2017 Association for Computational Linguistics

Lemma	POS	Inflection	Tag
luft	Noun	Nom. Pl.	NP
luft	Noun	Acc. Pl.	AP
luft	Noun	Gen. Pl.	GP
lüften	Verb	1^{st} Sg. Ind. Pres.	1SIE
lüften	Verb	1^{st} Sg. Subj. Pres.	1SKE
lüften	Verb	3^{rd} Sg. Subj. Pres.	3SKE
lüften	Verb	Sg. Imperative	RS

Table 1: An example of morphological analysis: multiple correct interpretations of the German word-form *lüfte*.

2.1 Alignment

For the training of the string transduction models, we need aligned source-target pairs. Monotonic alignments are inferred with a modified version of the M2M (*many-to-many*) aligner of Jiampojamarn et al. (2007), which maximizes the joint likelihood of the aligned source and target substring pairs using the Expectation-Maximization algorithm. A transduction from a word-form which happens to be shorter than its lemma (e.g. *lüfte/lüften*) could be achieved by including an insertion operation (e.g. $\epsilon \rightarrow n$). However, in order to avoid a prohibitively expensive transduction model, we model insertion as a many-to-many alignment, which bounds the transduction operation to its context.

We modify the M2M aligner by allowing the alignment to learn the likelihood of a generalized identity alignment (i.e., $i \rightarrow i$). Although inflection modifies some characters in a word, the majority of characters remain unchanged. This modification influences M2M towards small, single-character alignments.

The alignment of tags (e.g. 1SIE) merits special consideration. The tag is treated as a single indivisible unit, which is typically aligned to a substring in the word-form that involves the corresponding affix.[2] We allow the maximum length of the alignment substring to be longer for the tag alignment than for the individual characters in the lemma. After aligning the training data we record all substring alignments that involve affixes and tags. At test time, the source-target alignment is implied by the substring transduction sequence. We say that a lemma+tag analysis generated from a word-form satisfies the *affix-match constraint* if and only if the resulting affix-tag pair occurs in the alignment of the training data. Table 2 shows the alignments of five possible analyses to the corresponding word-form *schreibet*, of which three satisfy the affix-match constraint. Only analysis #2 (in bold) is correct.

s	c	h	r	e	i	b	et	
s	c	h	r	e	i	b	en+2PKA	✓
s	c	h	r	e	i	b	**en+2PKE**	✓
s	c	h	r	e	i	b	en+3SIA	×
s	c	h	r	e	i	b	en+3PIE	×
s	c	h	r	e	i	b	en+2PIA	✓

Table 2: Example alignments of hypothetical analyses of the German word-form *schreibet*. The check marks indicate which of the analyses satisfy the affix-match constraint.

2.2 Transduction

We train transduction models for transforming the word-forms into analyses on the aligned source-target pairs using a modified version of DIRECTL+ (Jiampojamarn et al., 2010). DIRECTL+ is a feature-rich, discriminative character transducer, which searches for a model-optimal sequence of character transformation rules for its input. The core of the engine is a dynamic programming algorithm capable of transducing many consecutive characters in a single operation, also known as a semi-Markov model. Using a structured version of the MIRA algorithm (McDonald et al., 2005), training attempts to assign weights to each feature so that its linear model separates the gold-standard derivation from all others in its search space.

DIRECTL+ uses a number of feature templates to assess the quality of a rule: source context, target n-gram, and joint n-gram features. Context features conjoin the rule with indicators for all source character n-grams within a fixed window of where the rule is being applied. Target n-grams provide indicators on target character sequences, describing the shape of the target as it is being produced, and may also be conjoined with our source context features. Joint n-grams build indicators on rule sequences, combining source and target context, and memorizing frequently-used rule patterns.

[2] Although our method can handle multiple tags, one tag is sufficient to represent the word-forms of the languages that we consider in this paper. The only exception is the circumfix of the German past participle.

Source	Target	
schreiben + 2PKA	schriebet	×
schreiben + 2PKE	**schreibet**	✓
schreiben + 3SIA	schrieb	×
schrieben + 2PKE	schriebet	×
schreiben + 2PIA	schriebt	×

Table 3: Example source-target pairs of the inflector model. The check marks indicate which of the analyses of the German word-form *schreibet* satisfy the mirror constraint.

Following Toutanova and Cherry (2009), we modify the out-of-the-box version of DirecTL+ by augmenting it with an abstract copy feature that indicates when a rule simply copies its source characters into the target, e.g. $b \rightarrow b$. The copy feature has the effect of biasing the transducer towards preserving the source characters during transduction.

In addition to training an *analyzer* model that transforms a word-form into an analysis, we also train an *inflector* model that converts an analysis back into a word-form. This opposite transformation corresponds to the task of morphological inflection (Cotterell et al., 2016). By deriving two complementary models from the same training set, we attempt to mimic the functionality of a genuine finite-state transducer. We say that a lemma+tag analysis generated by the analyzer model satisfies the *mirror constraint* if and only if the inflector model correctly reconstructs the original word-form from the analysis by returning it as its top-1 prediction. Table 3 shows five possible analyses of the word-form *schreibet*, of which only one satisfies the mirror constraint. Only analysis #2 (in bold) is correct.

2.3 Re-ranking

In order to produce multiple morphological analyses, we take advantage of the capability of DirecTL+ to output n-best lists of candidate target strings. To promote the most likely lemma+tag combinations, we re-rank the n-best lists using the Liblinear SVM tool (Fan et al., 2008), converting the classification task into the ranking task with the method of Joachims (2002).

The re-ranker employs several features, which are enumerated in Table 4. The first three features consider the form of the predicted lemma. Feature 1 indicates whether the lemma occurs at least once in a text corpus. Feature 2 is set to the normalized likelihood score of the lemma computed with a 4-gram character language model that is derived from the corpus. Feature 3 is the normalized confidence score assigned by DirecTL+.

	Description	Type
1	lemma in Corpus	binary
2	LM score	real
3	DirecTL+ score	real
4	affix match	binary
5	no affix match	binary
6	no affix match, top-1	binary
7	mirrored	binary
8	not mirrored	binary
9	not mirrored, top-1	binary

Table 4: Features of the re-ranker.

Features 4-6 refer to the *affix-match* constraint defined in Section 2.1, in order to promote analyses that involve correct tags. Features 4 and 5 are complementary and indicate whether the alignment between the affix of the given word-form and the tag of the predicted analysis was generated at least once in the training data. Feature 6 accounts for unusual affix-tag pairs that are unattested in the training data: it fires if the affix-match constraint in not satisfied but the analysis is deemed the most likely by DirecTL+.

Features 7-9 refer to the *mirror* constraint defined in Section 2.2, in order to promote analyses that the inflector model correctly transduces back into the initial word-form. These three features follow the same pattern as the affix-match features.

2.4 Thresholding

Each word-form has at least one analysis, but the number of correct analyses varies; for example, *lüfte* has seven (Table 1). The system needs to decide where to "draw the line" between the correct and incorrect analyses in its n-best list. Apart from the top-1 analysis, the candidate analyses are filtered by a pair of thresholds which are defined as percentages of the top analysis score. The thresholds aim at reconciling two types of syncretism: one that involves multiple inflections of the same lemma, and the other that involves inflections of different lemmas. The first threshold is unconditional: it allows any analysis with a sufficiently high score. The second, lower threshold is conditional: it only allows a relatively high-scoring analysis if its lemma occurs in one of the analyses that clear the first threshold. For example, the fourth analysis in Table 3, *schrieben* + 2PKE, needs to clear both thresholds, because its lemma differs from the top-1 analysis, *schreiben* + 2PKA. Both thresholds are tuned on a development set.

3 Experiments

In this section, we evaluate our morphological analyzer on English, German, Dutch, and Spanish, and compare our results to two other systems.

3.1 Data

We extract complete inflection tables for English, German, and Dutch from the CELEX lexical database (Baayen et al., 1995). The number of inflectional categories across verbs, nouns, and adjectives is 16, 50, and 24, respectively, in the three languages. However, in order to test whether an analyzer can handle arbitrary word-forms, the data is not separated according to distinct POS sets. For consistency, we ignore German noun capitalization.

The Spanish data is from Wiktionary inflection tables, as provided by Durrett and DeNero (2013). and includes 57 inflectional categories of Spanish verbs. We convert accented characters to their unaccented counterparts followed by a special symbol (e.g. *cantó* → `canto'`), with no loss of information.

The data is split into 80/10/10 train/dev/test sets; for Spanish, we use the same splits as Durrett and DeNero (2013). We eliminate duplicate identical word-forms from the test data, and hold out 20% of the development data to train the reranker. The training instances are randomly shuffled to eliminate potential biases.

For re-ranking, we extract word-form lists from the first one million lines of the November 2, 2015 Wikipedia dump for the given language, and derive our language models using the CMU Statistical Language Modeling Toolkit.[3]

3.2 Comparison to Morphisto

We first compare our German results against Morphisto (Zielinski and Simon, 2009), an FST analyzer. Beyond morphological analysis, Morphisto also performs some derivational analysis, converting compound segments back into lemmas. For a fair comparison, we exclude compounds from the test set. In addition, because the lexicon of Morphisto has a limited coverage, we report micro-averaged results in this section.

Table 5 shows that overall our system performs much better on the test sets than the hand-engineered Morphisto, which fails to analyze 43% of the word-forms in the test set. If we disregard the word-forms that Morphisto cannot handle, its F-score is actually higher: 89.5% vs. 84.0%.

3.3 Comparison to Marmot

Marmot (Müller et al., 2013) is a state-of-the-art, publicly available morphological tagger[4], augmented with a lemmatizing module (Müller et al., 2015), which can also take advantage of unannotated corpora. In order to make a fair comparison, we train Marmot on the same data as our system, with default parameters. Because Marmot is a morphological tagger, rather than an analyzer, we provide the training and test word-forms as single-word sentences. In addition, we have modified the source code to output a list of n-best analyses instead of a single best analysis. No additional re-ranking of the results is performed, as Marmot already contains its own module for leveraging a corpus, which is activated in these experiments. Separate thresholds for each language are tuned on the development sets. (c.f. Section 2.4).

Table 6 presents the results. We evaluate the systems using macro-averaged precision, recall, and F-score. Our system is consistently more accurate, improving the F-score on each of the four languages. Both systems make few mistakes on Spanish verbs.

The English results stand out, with Marmot achieving a higher recall at the cost of precision. English contains more syncretic forms than the other three languages: 3 different analyses per word-form on average in the test set, compared to 1.9, 1.3, and 1.1 for German, Dutch, and Spanish, respectively. Marmot's edit-tree method of candidate selection favors fewer lemmas, which allows the lemmatization module to run efficiently. On the other hand, DIRECTL+ has no bias towards lemmas or tags. This may be the reason of the substantial difference between the two systems on Dutch, where nearly a quarter of all syncretic test word-forms involve multiple lemmas.

An example of an incorrect analysis is provided

[3] *http://www.speech.cs.cmu.edu*

[4] *http://cistern.cis.lmu.de/marmot*

	English			German			Dutch			Spanish		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DirecTL+	**93.5**	88.9	**91.2**	**87.3**	**88.7**	**88.0**	**87.3**	**90.3**	**88.8**	**99.3**	**99.5**	**99.4**
Marmot	87.5	**94.3**	90.8	85.3	88.5	86.9	81.3	84.7	82.9	99.2	98.9	99.1

Table 6: Macro-averaged results on four languages.

System	P	R	F1
DirecTL+	**78.7**	**92.6**	**85.1**
Morphisto	65.1	52.7	58.2

Table 5: Micro-averaged results on German.

by Spanish *lacremos*. Both systems correctly identify it as a plural subjunctive form of the verb `lacrar`. However, Marmot also outputs an alternative analysis that involves a bizarre lemma `lacr`. Our system is able to exclude this wordform thanks to a low score from the character language model, which is taken into consideration by the re-ranker.

4 Conclusion

We have presented a transduction-based morphological analyzer that can be trained on plain inflection tables. Our system is highly accurate, and has a much higher coverage than a carefully-crafted FST analyzer. By eliminating the necessity of expert-annotated data, our approach may lead to the creation of analyzers for a wide variety of languages.

Acknowledgments

The authors thank Ryan Cotterell for his comments regarding the Marmot source code. This research was supported by the Natural Sciences and Engineering Research Council of Canada, and the Alberta Innovates Technology Futures.

References

Harald R. Baayen, Richard Piepenbrock, and Leon Gulikers. 1995. *The CELEX Lexical Database. Release 2 (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, Pennsylvania.

Grzegorz Chrupała, Georgiana Dinu, and Josef Van Genabith. 2008. Learning morphology with Morfette. In *LREC*.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared task: morphological reinflection. *ACL 2016*, page 10.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *HLT-NAACL*, pages 1185–1195.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion. In *NAACL-HLT*, pages 372–379.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2010. Integrating joint n-gram features into a discriminative training network. In *NAACL-HLT*.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *ACL*.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *EMNLP*, pages 322–332.

Thomas Müller, Ryan Cotterell, and Alexander Fraser. 2015. Joint lemmatization and morphological tagging with LEMMING. In *EMNLP*.

Tommi A. Pirinen. 2015. Omorfi - free and open source morphological lexical database for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania*, number 109 in NEALT Proceedings Series, pages 313–315. Linköping University Electronic Press, Linköpings universitet.

Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. 2009. A new hybrid dependency parser for german. *Proceedings of the German Society for Computational Linguistics and Language Technology*, 115:124.

Kristina Toutanova and Colin Cherry. 2009. A global model for joint lemmatization and part-of-speech prediction. In *ACL*, pages 486–494.

Andrea Zielinski and Christian Simon. 2009. Morphisto-an open source morphological analyzer for German. In *Finite-state Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP; Edited by Jakub Piskorski, Bruce Watson and Anssi Yli-Jyrä*, volume 191, page 224. IOS Press.