Semi-Automatic Extension of Morphological Lexica

Tobias Kaufmann and Beat Pfister
Speech Processing Group
Swiss Federal Institute of Technology (ETH)
Zurich, Switzerland
Email: {kaufmann,pfister}@tik.ee.ethz.ch

Abstract—We present a tool that facilitates the efficient extension of morphological lexica. The tool exploits information from a morphological lexicon, a morphological grammar and a text corpus to guide the acquisition process. In particular, it employs statistical models to analyze out-of-vocabulary words and predict lexical information. These models do not require any additional labeled data for training. Furthermore, they are based on generic features that are not specific to any particular language. This paper describes the general design of the tool and evaluates the accuracy of its machine learning components.

I. INTRODUCTION

MANY applications of natural language processing heavily rely on lexical resources. For highly inflected languages, such resources are typically represented as morphological lexica that contain stem entries, inflectional morphemes and derivational morphemes. The formation of words from morphemes is described by a morphological grammar.

In practical applications, lexical resources often need to be extended continually, either to include additional in-domain words or to expand the application to new domains. The manual extension of lexical resources is considered to be expensive. This is particularly true if lexical resources of different languages have to be maintained. In such cases, a trained native speaker may not be readily available.

One way to deal with this problem is to compromise on the quality of the lexicon. For example, out-of-vocabulary words can simply be added as full forms with some minimal specification, e.g. their part-of-speech. Such an approach is problematic for two reasons. First, the underspecification increases the ambiguity in later processing stages such as syntactic analysis or generation. Second, adding full forms is rather inefficient for highly inflected languages, where a single stem can account for many inflected forms and a potentially infinite number of compound words. The latter problem can be partially solved by stripping off the suffix of the out-of-vocabulary word (either manually or by means of heuristics) and allowing the resulting pseudo-stem to undergo arbitrary inflection. This, however, will again increase the ambiguity.

In this paper, we present a tool that facilitates the efficient extension of high-quality morphological lexica, even for a user with minimal knowledge of the underlying linguistic representations and conventions. The tool assists the user in creating the stem entry that is required to correctly analyze a given out-of-vocabulary word. The acquisition process is guided by statistical models whose training requires no additional data apart from a text corpus and some language-specific

parameters. The models are based on a set of generic features and thus can be trained for arbitrary languages.

The paper is structured as follows. Section II describes the concepts underlying our tool and illustrates a typical user interaction. Section III is concerned with the statistical models, their training procedures and their accuracy. Related work is reviewed in Section IV and Section V concludes with a final discussion.

II. TOOL DESIGN

A. Basic Assumptions

Before describing the actual tool, we want to make certain assumptions explicit. Regarding the linguistic processing, we assume that the morphological grammar is a context-free grammar with atomic-valued attributes attached to the nonterminal symbols. In this setup, a stem entry consists of an orthographic representation of the stem, a preterminal symbol and a value for each attribute. The attributes of the preterminal symbol denote syntactic features (e.g. grammatical gender for German nouns) or inflectional information such as the inflectional class. For example, the stem entry for the German noun *Tür* (*door*) can be defined as follows:

```
NS(sk10,pk4,f) "tür+"
```

The first two attributes denote the inflectional classes for singular and plural inflection, and the third attribute indicates feminine gender. The orthographic representation of the stem is terminated by the morpheme boundary marker +.

Morphological processing may also involve a finite-state transducer (FST). FSTs essentially map the lexical form of a morpheme to one or more surface forms depending on the graphemic context. For example, the surface form of the German verb *handeln* (*to trade*) can be specified as `handel+`. The FST ensures that this lexical form has the surface realization *handl* if followed by an *e* but *handel* anywhere else. Our tool can deal with FSTs by transforming surface forms back to possible lexical forms.

A final assumption is that an out-of-vocabulary word contains exactly one unknown stem. To put it differently, our tool will always look for a single stem that allows to analyze a given out-of-vocabulary word. If the word contains two or more unknown stems, the acquisition will result in a pseudo-stem which is itself a compound.

B. Acquisition Process

The acquisition process consists of three steps. In the first step, an out-of-vocabulary word is entered and automatically analyzed.

In the second step, the stem's preterminal symbol and lexical form are determined. To this end, the tool shows all possible stem hypotheses that would render the given out-of-vocabulary word analyzable if added to the lexicon. The visualization of the stem hypotheses reflects their relative probabilities, which allows the user to focus on the most likely hypotheses. The user can reduce the candidate set by adding constraints on various levels of linguistic abstraction. The second step is completed by selecting a stem hypothesis.

The morphosyntactic features of the stem entry are determined in the third step. The tool facilitates this task by providing a probability for each potential attribute value. Further, the tool offers different ways for verifying and correcting the predictions.

As soon as the selected feature values are confirmed, a new stem entry is generated and added to a special lexicon. The second and the third step will be illustrated and further detailed in the following sections.

C. Determining the Stem

Figure 1 shows the state of the dialog at the beginning of the second step. Panel 1 visualizes the out-of-vocabulary word under consideration (in the present example the German noun *Dressursport*, engl. *dressage sports*). The possible stem hypotheses are listed beneath this word in the order of their respective probability. The hypothesis on the top, the noun stem with lexical form `dress+`, is the most likely one. In order to help the user in verifying the top-ranked hypothesis, panel 3 lists the five most frequent words that can be analyzed with the respective stem.

In this example, the correct hypothesis (the noun stem `dressur+`) is ranked second. A user which is familiar with the given lexical representation can simply click on the correct stem and thereby proceed to the third step. If the user is less trained or if the correct stem is not among the best few hypotheses, the candidate set can be reduced in three ways.

1) Related words: Panel 5 lists words that can be analyzed with one of the available stem hypotheses *but not* with the top-ranked hypothesis. By clicking on a word, the candidate set is restricted to the stems which would allow to analyze that word. For example, clicking on *dressur* (*dressage*) results in a set of four stem hypotheses, the top-ranked hypothesis being the correct one. The word constraint is added to panel 4 and can be removed with another click. Note that this interaction requires no linguistic knowledge besides some sense of semantic relatedness.

The list of related words is constructed as follows. Starting with the second-ranked hypothesis, each stem hypothesis contributes up to three words that discriminate it from the top-ranked hypothesis. If there are more than three such words, only the three most frequent ones (in terms of corpus
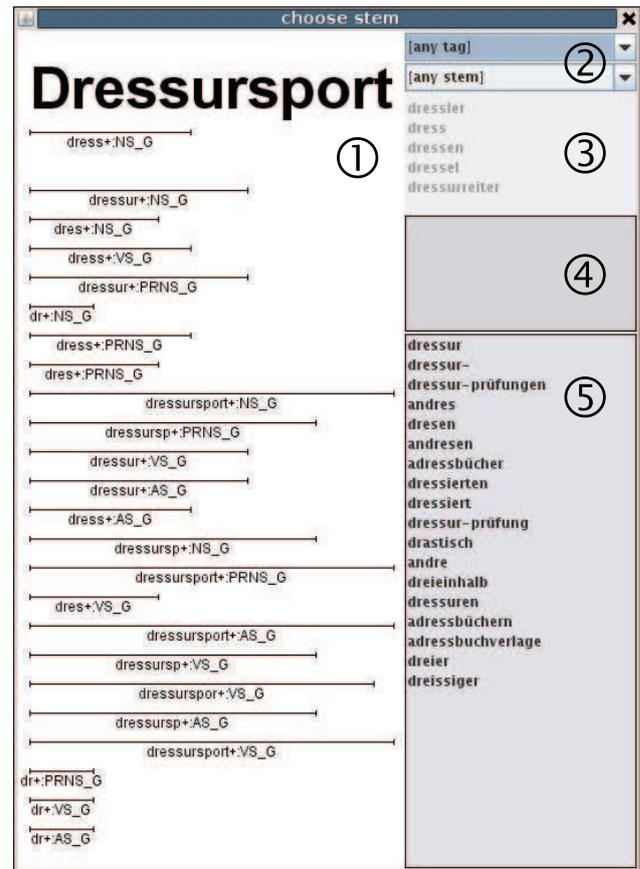

Fig. 1.   The dialog for determining the preterminal and the lexical form of the stem. The stem hypotheses shown on the left are ordered by probability, the topmost hypothesis being the most likely one.

occurrences) are considered. Thus, the words at the top of the list correspond to the most probable stem hypotheses.

2) Stem span: By clicking on a letter of the out-of-vocabulary word in panel 1, the user can specify whether this particular letter should be included in the stem or not. As for the related words, the candidate set is reduced to include only stem hypotheses that fulfill this constraint. For example, clicking on the second letter 'r' results in a set of 13 candidates. Again, the top-ranked hypothesis happens to be the correct one.

This kind of interaction requires the user to be familiar with the particular notion of a stem that is assumed in the morphological lexicon. This notion may not be intuitive. For example, the lexical stem may differ from the surface stem due to the FST, or parts of the "actual" stem may be represented by special morphemes in order to account for allomorphic variation.

3) Tag and preterminal: A final way of restricting the set of stem hypotheses is to specify the tag or the preterminal symbol in panel 2. The tag is the non-terminal symbol that represents the out-of-vocabulary word on the sentence level. Tags typically correspond to parts-of-speech such as noun, verb or adjective. Even though the tool is able to predict the
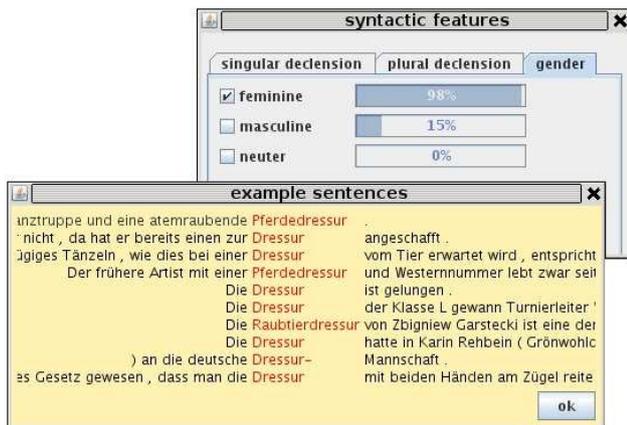
Fig. 2. The dialog for determining the morphosyntactic features. The window in front presents corpus examples which indicate that the noun stem *dressur* has feminine gender.

part-of-speech of a word (see Section III-B), this information is currently not used to restrict the candidate sets. The preterminal symbol denotes the category of the stem, e.g. noun stem, verb stem or adjective stem.

D. Determining the Morphosyntactic Features

The dialog for determining the morphosyntactic features is shown in Figure 2. There is a tab for each morphosyntactic feature. A tab displays information for each possible value of its corresponding feature. In particular, it indicates the probability that a certain value is present in the given stem. These probabilities are computed by binary classifiers (one classifier per value) and thus do not sum to 1. The use of binary classifiers is motivated by the fact that values need not be mutually exclusive. For example, the German noun *filter* (engl. *filter*) can have either masculine or neuter gender.

Initially, the most probable value of each tab is checked. If the user agrees with this choice, the tool can directly generate the final stem entry. If not, the selection can be changed by checking or unchecking individual values. The validation and correction of the automatically predicted values is assisted in two ways:

First, the tool can present corpus examples which suggest that a particular value should be chosen. Figure 2 shows 10 corpus examples supporting the hypothesis that *Dressur* is feminine (which it is). The most informative example is shown on the top of the list. In this example, the best indicator for feminine gender seems to be the unambiguously feminine determiner *eine* followed by a word that ends in *-e*. The prototypical female determiner *die* is less discriminative as it can also appear with plural nouns of any gender.

Corpus examples tend to be less useful for verifying inflectional features. For instance, if the best predictor for some inflectional class is the grammatical gender, the tool will present corpus examples which indicate that gender. This information clearly doesn't help the verification. Thus, the tool can visualize the current inflectional paradigm as tables and

lists of inflected forms. This layout of this visualization is defined as a grammar and language specific template that has to be created manually. The user can also introduce constraints by restricting the inflected forms for specific syntactic contexts. As a result, the incompatible values will be disabled in the dialog.

III. MACHINE LEARNING

The presented tool is based on machine learning components that compute the probability of a stem hypothesis and predict the morphosyntactic features. The underlying models are automatically created from a morphological lexicon, a morphological grammar, a normalized text corpus and some grammar-specific information. In the following, we will first comment on the latter two resources and then describe the main components. Finally, the accuracy of the models will be evaluated.

A. Resources

The most important additional resource is a sufficiently large text corpus. The corpus is assumed to be normalized such that there is exactly one token per line. A token is either a word or an interpunction symbol.

The tool also requires some explicit information about the given language and grammar. First of all, it needs a description of the stem morphemes that are to be acquired. Besides the preterminal symbol, a stem description specifies how each argument has to be handled: an argument can either be determined by the acquisition tool, set to a default value or remain unspecified. Next, the lexical form of a stem has to be described as a regular expression. The lexical form will typically be an arbitrary sequence of characters followed by a morpheme boundary symbol, e.g. `[a-zäöü]+[+]` in the case of our German grammar.

Finally, the explicit information also includes two sets of heuristics that allow for a very crude disambiguation. The first set relates a tag to its preferred heads. For example, German nouns can be derived from noun stems, adjective stems or verb stems, but the preferred head (in case of ambiguity) is the noun stem. The second set excludes certain tags for certain stems. For example, if an overly productive rule allows to create a proper name from each noun, it is hardly possible to train a part-of-speech tagger that can discriminate between these two. Thus, it can be declared that the proper name tag should be excluded if a word can be derived from a noun stem.

B. Part-of-Speech Tagging

Part-of-speech information can be applied on different levels of our approach. We have implemented an HMM part-of-speech tagger similar to the TnT tagger [1]. The tagger is based on a trigram sequence model and exploits suffix statistics for handling unknown words.

The tagger is trained on the text corpus by means of an expectation-maximization algorithm. For each analyzable word, the set of possible tags (i.e. non-terminal symbols) is defined by the grammar and the lexicon. We initially

assume that the possible tags are uniformly distributed. The tag distributions of out-of-vocabulary words are estimated from the suffix statistics of the analyzable words. In each training iteration, the forward-backward algorithm is employed to compute the posterior tag probabilities of each word. This information is then used to re-estimate the model parameters. As the forward-backward algorithm is rather slow with logarithmic probabilities, we preferred a scaling approach to avoid underflow [2], [3].

The tagger is used to produce a tagged version of the text corpus. Further, the tag distribution of each word is computed as the average tag distribution for all corpus occurrences.

C. Predicting Stems

Predicting stem entries is considered to be a discriminative reranking task. First, we compute all stem hypotheses that would render the given out-of-vocabulary word analyzable. Next, each stem hypothesis is transformed into a stem entry with unspecified arguments. These additional entries allow to analyze the out-of-vocabulary word w, yielding a set of parse trees $\mathcal{T}(w) = \{t_1, t_2, ..., t_n\}$. A disambiguation model then assigns a conditional probability $P(t|\mathcal{T}(w))$ to each parse tree. The probability of a stem s is defined as

$$\tilde{P}(s|\mathcal{T}(w)) = \max_{t \in \mathcal{T}(w,s)} P(t|\mathcal{T}(w)), \quad (1)$$

where $\mathcal{T}(w, s)$ denotes the set of those parse trees that include the hypothetical stem entry s[1]. The distribution $P(t|\mathcal{T}(w))$ is described by a discriminative log-linear model [6], [7]:

$$P(t|\mathcal{T}(w)) = \frac{\exp(\sum_i \theta_i f_i(t))}{\sum_{t' \in \mathcal{T}(w)} \exp(\sum_i \theta_i f_i(t'))} \quad (3)$$

The real-valued features $f_i(t)$ represent pieces of evidence whose relative importance is expressed by the feature weights θ_i. In the present work, each feature $f_i(t)$ counts how often a certain linguistic event occurs in the parse tree t.

Table I shows the different types of linguistic events that were considered for stem prediction. Actual linguistic events are defined by replacing the arguments X, x, c, M, $i^\star$ or m_i with specific values. The character classes that are used to characterize stem hypotheses are automatically induced from the text corpus with a clustering algorithm [8]. Both 8 and 16 classes are considered in order to capture different levels of abstraction. The stem occurrence information is extracted by analyzing the 500 000 most frequent words in the corpus.

[1] The correct way of computing the stem probability may be considered to be

$$P(s|\mathcal{T}(w)) = \sum_{t \in \mathcal{T}(w,s)} P(t|\mathcal{T}(w)). \quad (2)$$

This probability can be efficiently computed from a packed parse forest representation [4] without explicitly enumerating all parse trees. However, we observed that the prediction of the correct stem hypothesis was significantly less accurate for this approach. In particular, we observed a strong bias towards stems that are more productive in generating parse trees (i.e. verb stems in our German grammar). We believe that this is due to the training procedure, which can be interpreted as maximizing some margin between the correct parse tree and the incorrect ones [5].

TABLE I
TYPES OF LINGUISTIC EVENTS FOR STEM PREDICTION

stem hypothesis (STH) features
STH has lexical form x
STH covers at most m surface characters ($1 \leq m \leq 10$)
STH contains the lexical character sequence x ($|x| \leq 4$)
STH contains the character class sequence x ($|x| \leq 4$)
STH has lexical prefix x ($|x| \leq 4$)
STH has lexical suffix x ($|x| \leq 4$)
STH has lexical prefix x and preterminal X ($|x| \leq 4$)
STH has lexical suffix x and preterminal X ($|x| \leq 4$)
STH has lexical prefix x and left surface context c ($|x| \leq 4$, $|c| \leq 3$)
STH has lexical suffix x and right surface context c ($|x| \leq 4$, $|c| \leq 3$)
parse tree features
tree contains terminal X
tree contains non-terminal X
tree contains rule X
tree contains lexicon entry X
tree contains morpheme X
tree contains morpheme with lexical form x
tree contains morpheme with surface form x
tree contains morpheme with lexical form x and preterminal X
tree contains morpheme with surface form x and preterminal X
morpheme has lex. prefix x and left surf. context c ($|x| \leq 4$, $|c| \leq 3$)
morpheme has lex. suffix x and right surf. context c ($|x| \leq 4$, $|c| \leq 3$)
capitalization features
word is capitalized and the parse tree's root non-terminal is X
word is not capitalized and the parse tree's root non-terminal is X
stem occurrence features
the corpus contains a word that can be decomposed into the morpheme sequence $m_1, ..., m_M$, where $m_{i^\star}$ is the stem hypothesis. Each morpheme m_i is specified by its preterminal symbol and (optionally) its lexical form.

Estimating the model parameters θ_i requires training examples, each example consisting of a correct parse tree and a set of incorrect parse trees. As the number of parse trees can be huge, a training example is generated by means of random subsampling [9]: we randomly choose one tree that contains the correct stem and 20 trees that contain any other stem. After generating 10 000 training examples for different words w, Charniak and Johnson's MaxEnt reranker [7] is used to compute the model parameters. Note that the potentially very large number of features requires some kind of feature selection. For this purpose, we only consider the first 2000 features that are chosen by a boosting learner [10].

At this point, we still require a list of out-of-vocabulary words for which the correct stem hypothesis is known. As no manually labeled data is available, the lexicon and the text corpus are used to artificially generate such data. This process is described in the following.

An in-vocabulary word w from the text corpus is morphologically analyzed and disambiguated by means of a simple heuristics: a parse tree is preferred if (in decreasing order of importance) its root non-terminal matches the most likely tag of w according to the tagger, if it contains less stems, if it contains more stems that are preferred heads of the root tag (see Section III-A), if it contains longer stems and if it is smaller in terms of tree nodes.

From the resulting unambiguous parse tree, a random stem is selected. This stem will be considered as the correct stem hypothesis. Next, the stem is removed from the lexicon, and

TABLE II
FEATURE TYPES FOR PREDICTING MORPHOSYNTACTIC FEATURES

word at position P is x ($-2 \leq P \leq 2$)
word at position P has suffix x ($\|x\| \leq 4$, $-2 \leq P \leq 2$)
stem contains the lexical character sequence x ($\|x\| \leq 4$)
tag at position P is X ($-2 \leq P \leq 2$)

TABLE III
STEM PREDICTION: EVALUATION

approach	accuracy
baseline	8%
no stem occurrence features, unconstrained	53%
no stem occurrence features, constrained	57%
stem occurrence features, unconstrained	68%
stem occurrence features, constrained	72%

so are all overlapping stems from alternative parse trees. In general, w will now be out-of-vocabulary, which allows to generate the stem hypotheses. Given the word w, the correct stem hypothesis and the set of incorrect stem hypotheses, a training example for the log-linear model can now be generated.

In order to reduce the mismatch between the training data and the actual out-of-vocabulary words, each (correct) stem occurs about equally often in the training data. This prevents the model from rote learning the properties of a few very frequent stems.

D. Predicting Morphosyntactic Features

Predicting a morphosyntactic feature involves one classifier for each possible value of the given feature. In our German system, for example, there is a classifier which predicts whether a given stem has feminine gender or not.

The decision of the classifier is based on a set of boolean features. Each feature can be thought of as scanning the text corpus and reporting whether the stem under consideration occurs in a specific context. For example, some feature may be true if and only if the corpus contains a word that can be analyzed with the given stem and is preceded by the word *eine*. The basic feature types are shown in Table II. The positions indicated by P are relative to the given stem occurrence, i.e. the word a position -1 immediately precedes the stem occurrence.

The actual classification process is as follows. First, a pre-computed index is used to retrieve all corpus words which can be analysed as having the given stem as their rightmost stem. The rightmost stem is typically the head of a compound word. For languages with left-headed compounds (e.g. Vietnamese), the assumed head position can be changed accordingly. Next, a second index provides the positions where those words occur in the corpus. The aggregated value of a boolean feature is computed as the logical disjunction of the feature values for the different corpus occurrences.

The training of the classifiers is straight-forward: training examples can be generated by extracting the feature values for the known stems from the morphological lexicon, which also provide the correct label. As classifiers, we use maximum entropy models [11] with Gaussian priors for regularization [12]. We apply feature selection based on a mutual information criterion: the class label and the value of a particular feature can both be considered as random variables, where the random process consists of randomly choosing a stem occurrence in the corpus. The mutual information of these two random

variables provides a measure of the feature's usefulness. For each classifier, we select the 500 features with the highest mutual information.

In addition to the basic features from Table II, we also use so-called joint features. A joint feature is an arbitrary conjunction of (possibly negated) basic features. For example, a joint feature might indicate that the word at position -2 is *die* and the word at position -1 does not end in *-e*. Joint features are created automatically by training an alternating decision tree classifier [13] to predict whether some morphosyntactic feature has a specific value or not. Rather than using information aggregated over the whole corpus, this classifier makes a prediction for a single stem occurrence. Each decision in the alternating decision tree depends on the value of a single basic feature, and a path from the root node to a so-called predictor node corresponds to a joint feature. We used JBoost [14] to create a decision tree from 10 000 labeled stem occurrences. JBoost is run for 100 training iterations which results in 200 joint features.

E. Evaluation

In order to evaluate the accuracy of stem prediction, we randomly chose 1000 out-of-vocabulary words from our German text corpus (220 million tokens of newspaper text). Words with typographic errors, foreign words, dialect words and closed-class words were removed from this set. Also, personal names and geographical names were not considered because they are massively underrepresented in our German lexicon. As it is assumed that compounds do not contain two or more unknown stems, such compounds were excluded as well. Finally, we removed compounds which our German grammar could not account for. For the remaining 530 words, the correct stems were determined manually.

Table III shows the resulting accuracies. The baseline is the expected accuracy when uniformly choosing a random stem hypothesis. The baseline is compared to four variants of our stem prediction algorithm. These variants differ in whether they make use of stem occurrence features (see last item of Table I) and whether tag constraints are applied.

With tag constraints, the best out-of-vocabulary stems are not directly determined according to $\tilde{P}(s|\mathcal{T}(w))$. Rather, the maximization in equation (1) is additionally restricted to parse trees whose root non-terminal matches the most likely tag of w according to the part-of-speech tagger. If the tagger does not make a prediction for w, the algorithm falls back on $\tilde{P}$. Unlike the disambiguation model for stem prediction, the tagger also considers the textual contexts of a word and thus provides additional disambiguating information.

TABLE IV
PREDICTION OF MORPHOSYNTACTIC FEATURES: EVALUATION

gender	baseline	accuracy
masculine	55%	91%
feminine	62%	96%
neuter	79%	93%
singular declension class	baseline	accuracy
none	85%	98%
class 1	64%	94%
class 2	73%	89%
class 3	82%	90%
class 4	94%	95%
plural declension class	baseline	accuracy
none	80%	92%
class 1	94%	98%
class 3	80%	97%
class 4	80%	96%
class 6	91%	98%
class 7	74%	94%

The results show that the presented approach to stem prediction is fairly accurate: the most likely stem hypothesis is correct in almost 3 out of 4 cases. They also suggest that judging stem hypotheses on the basis of corpus occurrence information (rather than just considering the word w in isolation) leads to a much higher accuracy. Using part-of-speech tagging to disambiguate parse trees seems to have a smaller impact. We did not use this information in our tool in order to keep the user interface intuitive.

The prediction of morphosyntactic features was evaluated on about 300 noun stems that were excluded from training. Table IV shows the results for the different values of gender, singular inflection class and plural inflection class. For brevity, the table lists only the values that are observed in at least 5% of the stems. For each value, the baseline accuracy and the classifier accuracy are provided. The baseline accuracy is achieved by assigning each example the majority class. It can be seen that the classifier accuracies are substantially higher than the baseline accuracies. Exceptions are very rare values, for which a negative decision is already very accurate.

IV. RELATED WORK

There are a number of approaches to extracting lexicon entries from text corpora [15], [16], [17], [18]. Here, a lexicon entry consists of a lemma or stem and an inflectional paradigm, though some approaches also consider syntactic features [16], [18]. All approaches explicitly describe word formation, either in the form of a morphological grammar [16] or some representation of inflectional paradigms [15], [17], [18].

The above approaches do not consider existing morphological lexica in the acquisition process. Rather, they employ heuristics which are mainly based on the presence or absence of inflected forms in the corpus. For example, Oliver et al. prefer a hypothetical lexicon entry if it accounts for more inflected forms observed in the corpus [18]. Adolphs additionally considers the coverage of the paradigm, i.e. the fraction of generated inflected forms that actually appear in the corpus [16]. Šnajder et al. compute the corpus frequencies of the generated forms and essentially use the sum of these

frequencies as a measure of confidence. Finally, Forsberg relies on handcrafted constraints on the presence and absence of inflected forms [17].

Note that our approach also uses the presence of inflected form as evidence, both for predicting stems and for predicting morphosyntactic features. However, by exploiting the information in an existing morphological lexicon, we are able to compute a dedicated weight for each piece of evidence. Thus, inflected forms which are more reliable indicators have a higher influence on the prediction. The constraints used in Forsberg's approach [17] achieve a similar effect but have to be determined manually. Further, none of the above approaches attempts to analyze compounds: it is assumed that an unknown word is to be decomposed into a single stem and an inflectional ending.

Our approach to predicting morphosyntactic features is mainly related to some work on automatic lexical acquisition for precision grammars. Precision grammars require much richer lexical information than we are aiming at, e.g subcategorization frames. The most closely related work is that of Baldwin [19] and Nicholson et al. [20], who used k-nearest-neighbors classifiers to bootstrap lexical resources by means of corpus data. In contrast to the presented tool, they do not predict the stems of unknown words but assign lexical information to entire word forms. As in our approach, their features are based on word contexts, affixes and character n-grams. The former author additionally considers the output of a chunk parser and a dependency parser. We did not use such tools as they are not available for many languages.

V. DISCUSSION

We have presented a tool for the semi-automatic extension of morphological lexica. The tool facilitates the efficient acquisition of new stem entries from out-of-vocabulary words and ensures a high quality and consistency of the lexical resources.

1) *Efficiency*: Statistical models guide the user in the acquisition process and allow him to focus on the most promising hypotheses. The tool can also be handled by relatively untrained users: it employs concepts that are easily accessible to native speakers and its models capture some of the resource-specific "expert knowledge".

2) *Quality*: The user is offered different ways to verify his decisions, e.g. the predictions and the confidence of the statistical models, evidence from the text corpus and a visualization of the inflectional paradigm.

3) *Consistency*: Sometimes there are different possibilities to specify a stem entry. Sources of uncertainty may be the boundary between stem and affixes or the degree to which stems can include derivational morphemes. The statistical models can capture such tendencies in the original resources and may help to resolve ambiguities accordingly.

For our German resources, we have demonstrated that the machine learning components are able to predict lexical information with a reasonably high accuracy. We argue that the presented algorithms are applicable to a wide range of

languages. Apart from German, the tool has already been deployed for Swedish and Finnish[2], the latter being an agglutinative language which is not part of the Indo-European family.

ACKNOWLDEGEMENTS

REFERENCES

[1] T. Brants, "TnT: a statistical part-of-speech tagger," in *Proceedings of the sixth conference on Applied natural language processing*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 224–231.

[2] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[3] M. Johnson, "Why doesn't EM find good HMM POS-taggers," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 296–305.

[4] M. Tomita, *Generalized LR parsing*. Springer, 1991.

[5] S. Riezler, T. King, R. Kaplan, R. Crouch, J. Maxwell III, and M. Johnson, "Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques," *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pp. 271–278, 2002.

[6] M. Johnson, S. Geman, S. Canon, Z. Chi, and S. Riezler, "Estimators for stochastic "unification-based" grammars," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, Morristown, NJ, USA, 1999, pp. 535–541.

[7] E. Charniak and M. Johnson, "Coarse-to-fine n-best parsing and maxent discriminative reranking," *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pp. 173–180, 2005.

[8] P. Brown, R. Mercer, V. Della Pietra, and J. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[9] M. Osborne, "Estimation of stochastic attribute-value grammars using an informative sample," in *Proceedings of the 18th International Conference on Computational Linguistics*, 2000, pp. 586–592.

[10] M. Collins, "Discriminative reranking for natural language parsing," in *Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2000, pp. 175–182.

[11] A. L. Berger, V. J. Della Pietra, and S. A. Della Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[12] S. F. Chen and R. Rosenfeld, "A Gaussian prior for smoothing maximum entropy models," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep., 1999.

[13] Y. Freund and L. Mason, "The alternating decision tree learning algorithm," in *Machine learning: proceedings of the sixteenth international conference (ICML'99)*. Morgan Kaufmann Pub, 1999, p. 124.

[14] JBoost, http://jboost.sourceforge.net/index.html.

[15] J. Šnajder, B. Bašić, and M. Tadić, "Automatic acquisition of inflectional lexica for morphological normalisation," *Information Processing & Management*, vol. 44, no. 5, pp. 1720–1731, 2008.

[16] P. Adolphs, "Acquiring a poor man's inflectional lexicon for German," in *Proceedings of LREC'08*, 2008.

[17] M. Forsberg, H. Hammarström, and A. Ranta, "Morphological lexicon extraction from raw text data," in *Proceedings of FinTAL'06*. Springer, 2006, pp. 488–499.

[18] A. Oliver, I. Castellón, and L. Màrquez, "Use of internet for augmenting coverage in a lexical acquisition system from raw corpora," in *Proceedings of the International Workshop on Information Extraction for Slavonic and Other Central and Eastern European Languages*, 2003.

[19] T. Baldwin, "Bootstrapping deep lexical resources: Resources for courses," in *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, Ann Arbor, USA, 2005, pp. 67–76.

[20] J. Nicholson, T. Baldwin, and P. Blunsom, "Die Morphologie (f): Targeted lexical acquisition for languages other than English," in *Proceedings of the 2006 Australasian Language Technology Workshop*, 2006, pp. 67–74.

---

[2]In our evaluation for Finnish, the accuracy of stem prediction was observed to be around 45% for a relatively small corpus with about 15 million tokens.