Maximum Entropy Model for Disambiguation of Rich Morphological Tags

Mārcis Pinnis and Kārlis Goba

Tilde,
Vienibas 75a, LV-1004 Riga, Latvia
i pi i i g b ti e
http www ti e

Abstract. In this work we describe a statistical morphological tagger for Latvian, Lithuanian and Estonian languages based on morphological tag disambiguation. These languages have rich tagsets and very high rates of morphological ambiguity. We model distribution of possible tags with an exponential probabilistic model, which allows to select and use features from surrounding context. Results show significant improvement in error rates over the baseline, the same as the results for Czech. In comparison with the simplified parameter estimation method applied for Czech, we show that maximum entropy weight estimation achieves considerably better results.

Keywords: Tagger, maximum entropy, inflective languages, Estonian, Latvian, Lithuanian.

1 Introduction

The scope of this work covers three languages—Estonian, Latvian and Lithuanian, all of which have rich nominal and verbal morphology. While inflections in Estonian are formed agglutinatively, Latvian and Lithuanian share similar fusional morphology. All three languages exhibit high ambiguity of possible morphological analyses of a word, which in the case of Latvian and Lithuanian can be explained by their fusional nature, with several inflections sharing the same morphemes. In Estonian some agglutinative morphemes are shared between several inflections, producing homonymous surface forms.

1.1 Morphological Tagging

Morphological tagging can be viewed as a classification problem for a given word sequence (typically sentence), where each word is assigned a single tag describing its morphological properties. In this work, all three languages are processed within the same framework. Morphological analysis of a word (or in general, token) is encoded in a single tag consisting of fixed number of subtags corresponding to certain morphological categories (e.g., part of speech, gender, number, etc.).

Like in similar work for Czech [2], we take a two-step approach to tagging, where a token is first analyzed for possible morphological tags and disambiguated separately.

C. Mahlow and M. Piotrowski (Eds.): SFCM 2011, CCIS 100, pp. 14–22, 2011.

POS adjective
GENDER male
NUMBER plural
CASE nominative
DEGREE positive
DEFINITENESS indefinite

Fig. 1. Example of a morphological tag `p p` for the Latvian word **pašsaprotami** (lit. *self-evident*)

The morphological analyzer is based on a lemma lexicon and inflectional rules, and produces one or several analyses for a given word. The tagger then disambiguates the analysis by estimating probabilities of individual analyses and selecting the most probable.

In this work, we used an unified morphological analyzer consisting of a rule-based analysis module for Latvian and Lithuanian (developed by Tilde), and a separate analysis module for Estonian [6] (developed by Filosoft).

1.2 Morphological Tagset

The notion of tagset includes the set of valid combinations of subtags. Some subtags are mutually independent (e.g. a noun can decline in number and case independently), while others are valid only in certain contexts (e.g. tense is only valid for verbs).

The morphological tagset used for all three languages is similar to MULTEXT-East format [7] and consists of 28 categories. Each category is represented as a single-character subtag (see figure 1 for an example tag), with '0' corresponding to no value. While each language uses its own subset of all categories and their values, the category positions within the morphological tag and their meanings remain fixed.

2 Training Data

The training data (see figure 2 for a sample) for the morphological tagger consists of multiple lines; where each line represents a token and a sequence of possible tags. Sentences are separated with an empty line. The sequence of tags is given by the morphological analyzer of the particular language. The first tag is always the correct (manually annotated) tag. If the morphological analyzer does not recognize a token, it returns an empty tag. We assume that the morphological analyzer has recognized all tokens, thus the morphological tagger does not process unknown words and the tagging task is reduced to a morphological disambiguation task for known tokens.

We use morphologically disambiguated corpora for each of the three languages (Estonian, Latvian and Lithuanian) to train and test the morphological tagger.

Internal corpora were used for Latvian and Lithuanian, which consist of fiction, newspaper articles, scientific papers, business reports and letters, government documents, legal documents, student essays and theses, IT documents (such as manuals and web site information) and forum comments. Latvian and Lithuanian corpora were

```
 i      p           g
bi               i                              i          7
g                                          p
       t
```

Fig. 2. Latvian training data excerpt (lit. *all was at end*)

pre-tagged using a morphological analyzer and then given to annotators for manual disambiguation. Due to budget limitations, each token has been disambiguated only by one annotator, which lowers the corpus quality and creates unnecessary noise in the corpora.

For the Estonian tagger a freely available morphologically disambiguated corpus [9] was used, which consists of fiction, legal, newspaper and scientific texts. In this corpus, each word has been annotated by two annotators and disagreements have been resolved by a third annotator, thereby increasing the corpus quality. The Estonian corpus tagset is different to our unified tagset, therefore it had to be converted to the Multext-East tagset using a one-to-one transformation and a transformation from Multext-East to our unified tagset with some minor transformations to adjust the corpus to our unified morphological analyzer. In order to create the training data for the morphological tagger, the ambiguous tag sequence had to be created, therefore, the corpus was preprocessed also with our morphological analyzer.

After disambiguation, the corpora were split into training and test data so that none of the test sentences would be present in the training data. The final corpora statistics is shown in table 1.

Table 1. Training and test corpora

	Estonian	Latvian	Lithuanian
Total tokens	419,137	117,362	71,460
Sentences	31,266	6,564	4,201
Ambiguous words, %	32.4%	48.5%	36.0%
Word OOV rate	1.5%	3.0%	2.3%
Distinct tags	268	1401	1052
Tag perplexity	48.86	184.46	125.60
Test data, %	6%	10%	10%
Test tokens	26,366	12,826	8,103

2.1 Ambiguity Classes

Following the work for Czech [2], we use the notion of *ambiguity class* to describe possible morphological ambiguities within a subtag. For example, ambiguity class POS_{an} describes part of speech ambiguity between noun and adjective.

There are in total 216, 250 and 259 ambiguity classes throughout 22, 20 and 14 ambiguous morphological categories in the Latvian, Lithuanian and Estonian language training corpus respectively.

3 Model

The tagging model is based on the exponential probabilistic model used for Czech [2]. We assume that individual subtags $\{\mathbf{y}_{\text{POS}}, \mathbf{y}_{\text{TENSE}}, \mathbf{y}_{\text{GENDER}}, \ldots\}$ are independent, and model the probability of a candidate tag as a product of individual subtag probabilities:

$$p(\mathbf{y}) = \prod_{c \in \text{CAT}} p(\mathbf{y}_c) \,. \tag{1}$$

The subtag probabilities are modeled separately within each ambiguity class AC. The probability of an event y in context x is modeled as an exponentially weighted sum of feature functions [1]:

$$p_\Lambda(y|x) = \frac{\exp \sum_i \lambda_i f_i(y,x)}{Z(x)} \,, \tag{2}$$

where $f(y,x)$ are binary valued feature functions predicting event y in context x, and $Z(x)$ is the normalization factor. Here, events correspond to subtag values in a corresponding morphological category, and features describe the surrounding context of a word in a sentence.

4 Training

4.1 Feature Selection

The training of the morphological tagger heavily relies on the feature set used in the training and tagging process as can be seen in the results section. We use binary feature functions, which consist of a context address, function type (for instance, simple types, such as, part of speech, gender, number, also the token itself, or complex types, such as gender, number and case equality with the token whose category is being predicted) and the value of the function type (for example, '*a*' for part of speech or '*kas*' for a token in Latvian). We use the value ' ' to define equality of the function type of the token in the address defined by the function and the function type of the token whose category is being predicted. The first line of the feature excerpt in figure 3, therefore, is read in the following way: *if the next token is either a conjunction or a comma, the gender, number and case of the second token to the right have to agree with the gender, number and case of the predicted token*.

Our morphological tagger uses different feature sets for each of the ambiguity classes in the training corpus. Therefore, a feature selection algorithm was used in order to

```
R    e  e N  be    e
R
R
                       p
```

Fig. 3. First four feature excerpt for the Latvian part-of-speech ambiguity class '*qsv*'

select the best features that describe each of the ambiguity classes. But before the selection algorithm was applied, the initial feature set was generated using all possible categories, events, context position indicators (up to three tokens to the left and right) and some trigger words (conjunctions, prepositions, particles and adverbs) extracted from the training corpus. Although the trigger words increased the precision, the increase was very insignificant (in the order of 10^{-2} of a percent). This might be due to the fact that the part-of-speech feature functions already express the characteristics of the trigger words and, thus, the increase is very low. The feature generation resulted in 10017, 3801 and 3045 initial features for Estonian, Latvian and Lithuanian respectively.

When the initial feature set was created, a simple feature selection algorithm based on the maximal mutual information was used to select the set of feature functions with the highest score for each ambiguity class. The maximal mutual information of a feature function in an ambiguity class is

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} , \tag{3}$$

where $X = \{0,1\}$ corresponds to the binary value of feature function, Y is the set of possible events in the ambiguity class being processed (for instance, {'a', 'n'} for the ambiguity class 'an'), $p(x)$ is the probability of the feature function to receive the value x in the context of the ambiguity class, $p(y)$ is the probability of the event y in the ambiguity class and $p(x,y)$ is the probability of the feature function receiving the value x and the event simultaneously being y in the ambiguity class.

All probabilities are computed as normalized frequency distributions. Out of all initial feature functions a total of 1684, 775 and 742 feature functions were selected as important by the feature selection algorithm throughout all ambiguity classes for Estonian, Latvian and Lithuanian respectively for the best exponential models (applying a maximum of 150 feature functions in an ambiguity class for Estonian, 100 for Latvian and 50 for Lithuanian).

4.2 Model Parameters

We use a maximum entropy library developed at the Tsujii Laboratory of The University of Tokyo [8] to train the models of each of the ambiguity classes. The maximum entropy library features the LMVM (Limited Memory Variable Metric) parameter estimation [5], where parameter re-estimation, in comparison with iterative scaling algorithms, such as IIS (Improved Iterative Scaling) (for instance, in our tests IIS performed up to 30 times slower on the Latvian corpus using 150 features), converges significantly faster [4]. The estimated weights together with the feature sets of all ambiguity classes are combined in a single tagging model, which is used in the tagging process.

When disambiguating a token, we use the exponential model (1) to predict all events y in the context x for each ambiguity class of a token. Then we combine the probabilities of separate event predictions using a slightly modified version of the formula (1) for each possible tag [2]:

$$p(\mathbf{y}|x) = \prod_{c \in \mathrm{CAT}} (1-\alpha) p_{AC_c}(\mathbf{y}_c|x) + \alpha p_{AC_c}(\mathbf{y}_c) , \tag{4}$$

where we use linear interpolation of the model probability and the probability of the event *y* in the ambiguity class *AC* (which, in fact, is the frequency distribution of the event *y* in the ambiguity class *AC*) as a smoothing method. The α weights were manually estimated based on the highest training corpus precision. The usage of linear smoothing with the frequency distribution of an event in an ambiguity class has proven to increase the overall precision by 0.2–0.3%.

5 Results

We compare the error rates of the exponential model trained on Estonian, Latvian and Lithuanian data (table 2) with HunPos, a HMM trigram tagger [3]. We trained the exponential model with Maximum Entropy parameter estimation and the simplified parameter estimation described in [2]. The baseline error rate is computed using only the category label statistics (with $\alpha = 1$). HunPos tagger was run in guided mode, with possible morphological tags provided for each token.

We trained and evaluated the exponential maximum entropy models on various numbers of selected features (using the maximal mutual information feature selection method) and the best test results were achieved using 150, 100 and 50 features for Estonian, Latvian and Lithuanian respectively.

Table 2. Error rates

Experiment	Estonian	Latvian	Lithuanian
Baseline	9.72	14.00	7.47
HunPos	8.51	**6.67**	14.55
Exponential; simplified estimation	6.98	12.76	6.82
Exponential; ME estimation	**4.04**	8.49	**5.65**
Exponential; training data	3.07	5.32	3.76
Feature functions	150	100	50

Based on the best exponential maximum entropy models, we also evaluated the individual subtag error rates over all test tokens (table 3). The results suggest that for all languages the error rate distribution is fairly similar (with an exception of Estonian, in which gender is not used), more precisely, the categories with the most misclassifications are: part of speech, gender, number and case; case being the most difficult to predict.

5.1 Error Analysis

We have performed error analysis on the Estonian, Latvian and Lithuanian exponential models with 150, 100 and 50 feature functions respectively. For better interpretation of tagging errors, we grouped the errors by differences between the correct and the

Table 3. Error rates within categories

Category	Estonian	Latvian	Lithuanian
POS	2.11	1.91	2.33
GENDER	—	2.67	2.06
NUMBER	1.31	3.34	2.23
CASE	**2.15**	**4.53**	**2.64**
PERSON	0.29	0.37	0.37
TENSE	0.58	0.82	0.88
MODE	0.31	0.48	0.67
VOICE	0.60	0.51	0.63
REFLEX	—	0.05	0.09
NEGATIVE	0.30	0.00	0.05
DEGREE	0.50	0.80	1.01
DEFINITENESS	—	0.68	0.96
DIMINUTIVE	—	0.40	0.05
PREPNUMBER	—	0.60	—
PREPCASE	0.30	0.63	0.12
PREPTYPE	0.30	0.16	0.07
NUMTYPE	0.24	0.02	0.06
PRONCLASS	—	0.22	0.57
PARTTYPE	0.02	0.06	0.08
VERBTYPE	—	0.43	0.15
ADVTYPE	—	0.03	—
CONJTYPE	0.27	0.58	0.88

predicted tags. The cumulative error rates of the most common error types for each language (table 4) show that the top six errors cover approximately 50% of all errors in each language training corpus.

The error type, for instance, '*n* → *g (case)*' given in table 4 explains that instead of the case *n* (nominative) the case *g* (genitive) was selected as being more probable. Other error types in the table include number (*s* - singular, *p* plural), case (*p* - partitive, *a* - accusative) and part of speech (*a* - adjective, *c* - conjunction, *n* - noun, *q* - particle, *r* - adverb).

When analyzing the top six errors of the Latvian morphological tagger, it can be seen that the errors are fairly regular, for instance, for the error type '*m* → *f (gender)*' (as well as for the opposite) a common misclassification is done for the pronoun '*to*', which is obvious as the gender can either be distinguished by the sentence context (for instance, in noun phrases), by an anaphora resolution or cannot be distinguished at all in the case when the context is too small. As the feature functions do not consider anaphora resolution for pronouns of this type and the context may not reveal the correct gender, the statistical morphological tagger makes misclassifications. Another common misclassification occurs in noun phrases where adjuncts are used, for instance, consider the error type '*sa* → *pg (number & case)*'. The adjunct number and case in most cases, when observing the context to the right, can be identified, but the tagger makes a

Table 4. Top six error types

Language	Correct → Wrong (Category)	Error Coverage
Estonian	n → g (case)	13.22
	g → n (case)	24.85
	p → s (number)	32.25
	s → p (number)	39.36
	p → g (case)	45.01
	r → c (part of speech)	49.52
Latvian	pg → sa (number & case)	14.53
	m → f (gender)	26.31
	f → m (gender)	33.68
	sa → pg (number & case)	39.83
	pn → sg (number & case)	45.96
	a → n (case)	50.86
Lithuanian	pn → sg (number & case)	14.89
	f → m (gender)	28.16
	m → f (gender)	34.12
	q → c (part of speech)	39.08
	sg → pn (number & case)	44.01
	a → n (part of speech)	48.53

misclassification. This suggests that for specific ambiguity classes, either wrong feature functions have been prioritized or more complex feature functions would have to be generated, that address the issue of misclassification.

6 Conclusions

The results of the application of maximum entropy modeling to Estonian, Latvian and Lithuanian confirms the suitability of this method for morphologically rich languages and corresponds well to the results for Czech [2]. The exponential tagger performs significantly better than the baseline and in two cases significantly better than HMM tagger. In the case of Latvian, we have observed an interesting deviation in favor of HMM tagger. Also the high tagset perplexity for Latvian indicates that careful investigation of training data quality is necessary.

The feature selection algorithm used in our training and evaluation experiments does not consider interfeature relations, which lowers the final tagging precision because features, which in combination perform well, may not be selected and features, which in combination perform poorly, on the contrary, may be selected. Therefore, a better feature selection algorithm would be the use of iterative feature selection as explained by [2]. As we use the maximum entropy training method, the iterative feature selection would require large computing resources. An interesting experiment would be to run the iterative feature selection based on the simplified weight estimation algorithm and compare the results to the model acquired by maximum entropy training on the features selected by the iterative feature selection.

The tagger model could be extended to handle unknown words, allowing to avoid the shortcomings of the lexicon-based analyzer. In this case, the ambiguity class is unknown, and the model needs to be adjusted. One possibility would be to combine subtag classifiers trained on whole data (as opposed to conditioned by ambiguity class). In this case, some model of valid subtag combinations should be used to avoid predicting invalid tags.

The combination of subtag models currently treats all subtags equally. This combination could be parameterized by weighing the individual subtag probabilities in a log-linear fashion, effectively treating subtag probabilities as feature values. This approach would allow the parameters to be tuned and allow minimum error rate training. Also, more features (like subtag classifiers over all training data) could be added.

Acknowledgements. The research within the project Accurat leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007–2013), grant agreement no 248347.

References

1. Berger, A., Della Pietra, S., Della Pietra, V.: A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics 22(1), 39–71 (1996)
2. Hajič, J., Vidová-Hladká, B.: Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In: Proceedings of the COLING-ACL Conference, Montreal, Canada, pp. 483–490 (1998)
3. Halácsy, P., Kornai, A., Oravec, C.: HunPos — an Open Source Trigram Tagger. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 209–212. Association for Computational Linguistics, Prague (2007)
4. Malouf, R.: A Comparison of Algorithms for Maximum Entropy Parameter Estimation. In: Proceedings of CoNLL 2002, pp. 49–55 (2002)
5. Benson, S., More, J.: A Limited Memory Variable Metric Method in Subspaces and Bound Constrained Optimization Problems. In: Technical Report ANL/MCS-P909-0901, Argonne National Laboratory (2001)
6. Kaalep, H.-J.: An Estonian Morphological Analyser and the Impact of a Corpus on its Development. Computers and Humanities 31, 115–133 (1997)
7. MULTEXT-East: Multilingual Text Tools and Corpora for Central and Eastern European Languages, http i i M
8. A Simple C++ Library for Maximum Entropy Classification, http www t ii i t y p t e t
9. Morphologically Disambiguated Estonian Corpus, http www t ee p e p