Issues in Training the TreeTagger for Georgian

Dr Sophiko Daraselia

University of Leeds

Abstract

The paper describes the process of retraining the TreeTagger program (Schmid, 1994) for the Georgian language. This includes some general procedures such as designing a training corpus, creating a tagging lexicon, and training the TreeTagger on Georgian texts. I use a novel KATAG tagset and enclitic tokenization approach in part-of-speech tagging. The KATAG tagset is based on a new morphosyntactic language model (Daraselia and Hardie, fc).

In this paper, I address some major disambiguation issues that were revealed in training the TreeTagger on Georgian texts. I discuss some ways to get around these issues, such as implementing a workaround to the tagging lexicon. I report on the performance of the TreeTagger program and compare how different parameters such as the size of the training lexicon or context and affix lengths have effects on the Tagger's performance.

1. Introduction

The Georgian (*Kartuli*) language is an official language of Georgia and belongs to the Kartvelian language family[1]. Georgian has a complex and rich morphology, which presents interesting and challenging tasks in part-of-speech tagging (Daraselia and Hardie, fc).

There are multiple annotation systems for Georgian. Meurer (2007) describes a parser based on Lexical Functional Grammar (Kaplan & Bresnan 1982). This was used to tag the Georgian National Corpus[2]. The parser covers major POS and morphosyntactic features in addition to lexical and syntactic features such as verbs' transitivity. Thus, each token analysis is a collection of tags for specific features. Lobzhanidze's (2013, 2022) morphological analyzer is similar to Meurer's system which also annotates features such as verb's transitivity and noun animacy in addition to POS.

Unlike other schemas for Georgian described above (Meurer 2007, Lobzhanidze, 2013, 2022), KATAG operates fully at POS level and does not attempt complete morphological analysis. This is because just-POS tags have their own known uses and advantages to tagging morphosyntax alone. For instance, POS tagging categorises word tokens rather than applying collections of feature-tags. This means that tagged text with complex feature collections per word may require specialised interfaces, whereas querying one-analysis-per-token tagging is a standard concordancer function. Another advantage to tagging with one-analysis-per-token relates to unknown words. Based on form and context, a POS tagger can normally generate a guess at such words' category whereas feature-based systems may leave them unanalysed.

The main goal of part-of-speech tagging is to allow corpus search and statistics to be applied to single token-level grammatical categories. The TreeTagger program annotates for part-of-

[1] The Kartvelian language family includes the following four Kartvelian languages: *Georgian*, *Laz*, *Megrelian,* and *Svan*.

[2] The Georgian National Corpus is available at: http://gnc.gov.ge/gnc

speech and lemma information, and it is built into many widely used corpus query systems, such as #Lancsbox, CQPWeb, and TagAnt. The TreeTager trained on Georgian texts means that it is accessible in these environments.

I use a new KATAG tagset and enclitic tokenization approach. The KATAG tagset is based on a new morphosyntactic language. The design of KATAG is itself not treated in this paper, whose scope is purely its implementation using the TreeTagger program. The forthcoming paper on tagset-focused grammar modeling deals with this at length (Hardie and Daraselia, fc)[3] and this paper is not going to duplicate that. I use both part-of-speech tagging and lemmatization facilities[4] of the TreeTagger. However, the latter is not discussed in this paper, as the primary goal is tagging with KATAG.

2. General Procedures: Preliminary Tasks

2.1 KATAG Tagset: a brief overview

The KATAG is a fully hierarchical-decomposable tagset (Daraselia and Hardie, fc.). The components of the tags are mnemonics for the linguistic labels of the categories. Most mnemonic codes follow the conventions of the CLAWS tagsets for English. The hierarchy is represented left-to-right, with major POS first, sub-classifications, and morphological features. For example, *shen* 'you', receives the TAG *PP2SN* to signify that it is a *pronoun*, a *personal pronoun*, *second* person, *singular* in number, and in *nominative-absolutive* case.

Category	Number of Tags in KATAG tagset	Actual number of tags used in the training corpus
Verb	209	132
Pronouns	163	86
Numeral	58	44
Noun	24	8
Adjective	24	11
Particle	6	1
Residual	6	6
Punctuation	4	1
Adverb	3	3
Conjunction	2	2
Interjection	1	1
Postposition	1	1
Copula	1	1

Table 1: Number of tags in the KATAG tagset

The KATAG tagset consists of 502 tags in total, but only 219 tags (out of 502) appear in the training corpus and are used in POS-tagging. This means that 283 tags never appear in the

[3] Two forthcoming papers deal with the design and grammar model of the tagset:
(1) Hardie, A. and Daraselia, S. *A theory for words in Georgian: traditional constructs versus demands of corpus annotation.*
(2) Daraselia, S. and Hardie, A. *The KATAG morphosyntactic annotation schema for Georgian.*

[4] The trained TreeTagger program for Georgian performs POS-tagging and lemmatization. The KATAG tagset, parameter files, and fullform lexicon with lemma information are available on GitHub. In addition to this, #Lancsbox allows search/analysis according to type, POS, or lemma.

training corpus. These are the tags for some verbs and nominals (less frequent/rare use), such as suffixaufnahme cases in numerals (Daraselia and Hardie, fc).

Thus, the KATAG tagset includes information on major word classes, subclassifications, and morphology. Every word token receives exactly one tag, with clitics tagged separately from the word they are attached to.

2.2 Tokenization

The KATAG tagset requires certain clitics to be tokenized separately from their host words. It is implemented in TreeTagger using a rule-based enclitic tokenization program developed for this purpose. The rationale for this is beyond the scope of the present paper. These are dealt with at length in the tagset-focused grammar model paper (Hardie and Daraselia fc).

The TreeTagger provides several options for tokenization. It allows using (1) 'built-in' tokenization and (2) your own tokenization program. For this task, I designed my own rule-based enclitic tokenization program[5] which was applied prior to the in-built tokenization of the TreeTagger program. The built-in tokenization in the TreeTagger performs some general tokenization operations such as splitting tokens at a space, separating off punctuation, and then writing each token or punctuation to a separate line.

In part-of-speech tagging, I use a rule-based enclitic tokenization approach, where enclitic parts (enclitic postpositions and particles) of a word are tagged as units of their own. In general, tokenization is not a difficult task in Georgian as word breaks are clearly marked by means of spaces. In part-of-speech tagging (as opposed to morphological annotation, for instance) an affix does not receive its own tag but may affect the grammatical features marked on the word of which it is a part. Clitic receives its own tag, for example, the postposition *-ze* 'on' in Georgian. In order to achieve this, it must be tokenized as a unit of its own, separate from the host word to which it is phonetically and/or orthographically attached, even if this involves splitting up what might be considered 'one word'.

2.3 Disambiguation

Disambiguation is one of the most problematic tasks in part-of-speech tagging. I distinguish four different types of ambiguity in Georgian (cf. Cloeren, 1999: 47) including 1) *grammatical homonymy*, where one wordform isolated from its context belongs to more than one grammatical class; 2) *Categories without clear boundaries*, such as adjectives and adverbs in Georgian; 3) *genuine textual ambiguities*, where the context does not provide enough information for disambiguation and 4) *morphological syncretism*, where one wordform belongs to a specific morphosyntactic category, but it is difficult to identify appropriate morphosyntactic features, such as tense or argument agreement in verbs.

In the case of *grammatical homonymy*, one wordform belongs to more than one grammatical class when it is isolated from its context. For example, *dats'era* has two readings:

(1) Verbal noun ('to write')
(2) Verb ('S/he wrote it')

[5] The rule-based enclitic tokenization I developed is available on GitHub.

There are no clear boundaries for some categories in Georgian, such as adjectives and adverbs. For example, *lamazad* ‘beautifully’ can be analysed as an adjective in the adverbial case or as an adverb. Thus, it is difficult for a human annotator to decide on a single tag.

Cloeren (1999, p. 48) also describes *genuine* textual ambiguities, where text does not provide enough information for disambiguation. In Georgian, there are some genuine textual ambiguities such as the exclamatory word *smena* “listen/to listen”, where is unclear whether it is a verb or a noun.

The Georgian language has an additional level of ambiguity of *morphological syncretism*, when one wordform belongs to the same morphosyntactic category, but it is difficult to identify appropriate morphosyntactic features, such as tense and argument agreement in verbs. For example, the verb *gadzlevt* can have four different readings:

(1) ‘*S/he gives this to you (PL)*’. It is a verb, third person subject, singular in number and second person object, Plural in number.

(2) ‘*I give this to you (PL)*’. It is a verb, first person subject, singular in number and second person object, plural in number.

(3) ‘*We give this to you (PL)*’. It is a verb, first person subject plural in number and second person object plural in number.

(4) ‘*We give this to you (SG)*’. It is a verb, first person subject, plural in number and second person object, singular in number.

Some decisions have been made at the tagset design stage to address each of these disambiguation issues described above. For example, in the case of grammatical homonymy and morphological syncretism, ambiguous words were assigned two or more tags in the POS tagging lexicon and they were manually disambiguated in the training corpus. As for the words with no clear boundaries between the categories, a POS tagging lexicon has been developed for such categories. This means that if a word appears in the adjective lexicon, but in the text, it functions as a noun, it still will be tagged as an adjective. This was a consistent approach throughout the POS-tagging process.

3. Training the TreeTagger

3.1 Training corpus, tagging lexicon, and test set

Automated part-of-speech tagging presupposes manual tagging, which is needed as training data and is necessary for many computational part-of-speech tagging methods. As a training corpus, I use manually tagged text samples from the KaWaC corpus[6] (Daraselia and Sharoff, 2015). This includes randomly selected text samples from the corpus consisting of about 100,000 wordforms in total.

The training corpus was initially split into 80% in the training set, 10% in the development set, and 10% in the test set. At the POS-tagging stage, it was revealed that the training corpus was not sufficient to contribute to the tagging lexicon (also referred to as a Fullform lexicon). This was because the training corpus had a low number of unique token counts (about 8,000 unique tokens in a 100,000-word training corpus). For this reason, the tagging lexicon was enriched from other sources such as the Georgian monolingual dictionary and Georgian dialect dictionaries.

[6] KaWaC corpus is available at: http://corpus.leeds.ac.uk/internet.html

Training corpus	**Tagging (fullform) lexicon**	**Test set**
100,000 tokens	95,000 tokens from a range of sources: • 35,000 tokens from the KaWaC corpus • 40,000 words (lemma) from the Georgian Monolingual Dictionary (1950-1964) • 20,000 tokens from the Georgian dialect dictionaries	• 10,000 tokens from the training corpus • Additional 5,000 tokens from other sources

Table 2: Size of the training corpus, tagging (fullform) lexicon, and Test set.

The test set was randomly selected from the training corpus (10% split). The problem with the training corpus was that it came from the KaWaC corpus which primarily represents the press/news genre of the Georgian language. The trained TreeTagger demonstrated higher performance on this test set (see §5). The main goal was to make the tagger program easily adaptable for a wide range of inputs, not only for a particular genre of the language. For this reason, an additional, second test set was created from a range of genres. The second test set includes about 5,000 tokens including punctuation and other symbols. The training corpus and tagging (fullform) lexicon were used to create TreeTagger parameter files for automatic part-of-speech tagging of Georgian texts using the TreeTagger training program (Schmid, 1994). The TrainTreeTagger program requires the following datasets: a *fullform lexicon*, a *training set,* and an *open class list*. A fullform lexicon, also referred to as a tagging lexicon, contains the word form itself followed by a Tab character and a sequence of tag-lemma pairs. The tags and lemmata are separated by whitespace.

Token	**Tag**	**Lemma**
ადგილი	NSN	ადგილი
ადგილის	NSG	ადგილი
ადგილობრივ	JSU	ადგილობრივი
ადგილობრივებმა	JPE	ადგილობრივი
ადგილობრივი	JSN	ადგილობრივი
ადგილობრივმა	JSE	ადგილობრივი
ადგილს	NSD	ადგილი

Table 3: Example entries from the fullform lexicon.

The training set file contains the tagged training corpus in one-word-per-line format. This means that each line contains one token and one tag in that order separated by a tabulator.

Token	**Tag**
ამ	PDSE
ადამიანებმა	NPE
არაფერი	PNN
იცოდნენ	V:3P:I
პროგრამის	NSG
ფარგლებ	NPD
ში	II
ჩასატარებელი	JSN
საქმიანობის	NSG

თარბაზე	RR
.	YF

Table 4: Example data from the Training set.

The open class file contains a list of open class tags, i.e. possible tags of unknown word forms. This information is necessary to estimate likely tags of unknown words. The list covers six open class categories, such as adjectives, nouns, and verbs. The open class file contains 133 tags in total.

Open class category	Number of tags	Example
Verb	86	V:1P:A
Noun	17	NSE
Adjective	13	JSA
Numeral	11	MOSD
Pronoun	1	PIPD
Residual	5	FE

Table 5: Open class tags.

Thus, the TreeTagger was trained on the disambiguated and annotated training corpus containing about 100,000 wordforms. The tagging (fullform) lexicon contains the data from the Georgian monolingual dictionary and dialect dictionaries in addition to the training corpus.

4. Issues

4.1 Tagging Lexicon

In terms of Georgian morphosyntax, there is the issue of analysing the noun before a postposition: dative for function or nominative for form. This is dealt with at length in the tagset paper (Daraselia and Hardie, fc). In terms of the TreeTagger, I discuss how this is problematic in training the TreeTagger for Georgian.

Some major disambiguation issues were revealed in training the TreeTagger on Georgian texts. Specifically, these issues are related to the automatically derived suffix lexicon due to enclitic tokenization. I will illustrate this with the example *masalaze* 'on the material', where *ze* signifies the postposition 'on' and it is tagged separately. The problem here is that after decliticization the remaining word-form *masala* 'material' is ambiguous and can have three different readings and receive three different tags as follows: *NSN* (for noun, singular, nominative-absolutive) or *NSD* (for noun, singular, dative-accusative) or *NSA* (for noun, singular, adverbial) depending on the following postposition. The enclitic postposition -*ze* 'on' governs dative-accusative case in Georgian meaning that it should receive the *NSD* (for noun, singular, dative-accusative) tag.

In the fullform lexicon, ambiguous words have several possible tags. The tag order is given based on the wordform and its case order as defined in the tagging guidelines (Daraselia, 2019). The problem with the TreeTagger is that it is not designed to disambiguate such cases and assigns the most probable tag from the fullform lexicon. In this case, the word-form *masala* 'material' receives an *NSN* tag (the correct tag is *NSD*) incorrectly disregarding the postposition following it. This is because the high lexical probability of the nominative-absolutive tag always overweighs the contextual probability of dative-accusative before certain postpositions when the probabilities are combined together. This means that the tags in the fullform lexicon with high probability simply 'overrule' the disambiguation 'rule' (in the training corpus) of nominal+postposition pairs, and such wordforms always receive the most probable tag according to the fullform lexicon.

I proposed a way to implement a workaround to the fullform lexicon to avoid this problem by simply reducing the high probability of nominative-absolutive tags in the lexicon. Hereby, I introduce the term ‘normalization’ which refers to reducing high probabilities of certain tags in a way that avoids this problem. Specifically, I removed certain tags (such as tags for nominative-absolutive cases) from the fullform lexicon to ‘normalize’ their frequencies and probabilities and move them to an auxiliary lexicon. The purpose of the auxiliary lexicon is to select the correct tag when/if the wordform appears in the auxilary lexicon. Unlike the tagging lexicon, the auxiliary lexicon does not define probabilities of unknown words. By implementing this workaround, I avoid this problem. The removed wordforms are still used in POS tagging as an auxiliary lexicon. Including an auxiliary lexicon in POS tagging led to a significant performance increase. The normalized fullform lexicon, with the auxiliary lexicon, improved the TreeTagger performance. As a result of this, the TreeTagger successfully disambiguated over 98% of the nominals followed by a postposition.

This was one of the major problems with the TreeTagger, which is not really designed for this type of disambiguation task. The proposed workaround is the only way to do this in the context of the TreeTagger. Other more sophisticated techniques, such as rule-based approaches are likely to do better.

4.2 Wordform endings

The other major issue with the TreeTagger concerns the wordform endings. The TreeTagger program uses the suffix tree, where ‘the nodes are annotated with the letters of the word suffix in reversed order’ (Schmid, 1994). This is a major issue with the TreeTagger considering the complex morphology and agglutinative features of the Georgian language. The TreeTagger fails to distinguish between the major POS, such as a verb or noun due to ambiguous word endings. For example, *-bis* ending is either for verbs in present or future tense, or for nominals in genitive or dative-accusative (see §5).

Thus, this is an overall problem for the TreeTagger approach based on (short) beginning/end strings, which relies on unknown forms having suffixes similar to equivalent known forms that frequently occur. The suffixes that get learned by this procedure are much less likely to be useful for a morphologically rich language such as Georgian. Alternative approaches (such as regular expressions that identifies the presence of affixes using regular expressions rather than ‘ends-with’ rule; morphological analysis; paradigmatic lex memory) are likely to do better, but this cannot be done in the context of TreeTagger.

5. Performance and error analysis

5.1 Results

I tested several variations of the TreeTagger program by applying different values for parameters such as n-gram length and length of the suffix lexicon (Schmid, 1994). The best results were obtained within the following default values of the parameters of the TreeTagger:

- Minimum decision tree gain: 0.7
- Equivalence class weight: 0.15
- Minimum affix tree gain: 1.2
- Threshold probability for lexical entries: 0.001

Context	Prefix lexicon	Suffix lexicon	No of nodes	Max. pass length
Bigram	37 nodes	206 nodes	57	15
Trigram	37 nodes	206 nodes	85	16
Quatrogram	37 nodes	206 nodes	101	16

Table 6: Number of n-grams, affix nodes and the depth of the tree

In the first variation, zero frequencies are used and in the second variation, zero frequencies are replaced by 0.1 before the tag probabilities, to see how strong the influence of the choice of this parameter on the tagging accuracy is. However, changing the replacement value for zero frequencies in the decision tree from a very small value to 0.1 did not improve the accuracy. In both variations, the TreeTagger achieved an accuracy[7] of 88.45% in enclitic tokenization.

The inclusion of the auxiliary lexicon (85,000 words) initially dropped the accuracy of the TreeTagger to below 70% (initial accuracy without auxiliary lexicon is 88.45%). This is because the auxiliary lexicon did not include tags for ambiguous categories. This problem was easily solved by adding tags for ambiguous wordforms in the lexicon. Overall, the auxiliary lexicon has significantly increased the tagger's performance.

Method	Context	Accuracy
TreeTagger	bigram	88.45%
TreeTagger (with auxiliary lexicon)	bigram	92.41 %
TreeTagger (with auxiliary lexicon)	trigram	92.41 %
TreeTagger (withauxiliary lexicon)	quatrogram	92.41 %

Table 7: Comparison of accuracy of the TreeTagger program

Finally, the influence of the pruning threshold on the accuracy of the trigram version and the quatrogram version of the TreeTagger was tested. As shown in Table 7 above, increasing the context to trigram and quatrograms did not result in any improvement.

As a result of manipulation and normalization of the fullform lexicon, the TreeTagger achieved an accuracy of 92.41 %. Adding 3 or 4 tokens of context made no difference. Thus, the main contribution came from a better lexicon rather than longer contexts. This means that the quality of the human expert input is very important.

5.2 Error Analysis

The TreeTagger performance was tested on two test sets. The first test set (10,000 words) came from the training corpus (10% split) and the second test set was collected from the range of genres (5,000 tokens). In the first test set, the TreeTagger demonstrated higher performance (95%) than in the second test set (92.41%). This is because the training corpus primarily contains the press/news genre of the language and since the TreeTagger was trained on this training corpus, better results are obtained in this genre.

Tagging errors in part-of-speech categories vary greatly.

Part-of-speech	Proportion of the complete	Relative error	Coverage

[7] Accuracy (also known as 'correctness') here is defined as follows: percentage of correctly tagged tokens, divided by the total number of tokens (see van Halteren, 1999, p. 82).

	number of errors		
Verbs	3.89	34.05	11.43
Nouns	2.43	8.28	29.44
Adjectives	0.73	6.71	10.98
Pronouns	0.14	2.05	6.96
Numerals	0.1	3.35	3.05
Residuals	0.2	22.22	0.94

Table 8: Incorrectly assigned POS tags.

The tagger performance is evaluated for both known and unknown words. Overall, 19.03% of the words in the test set are unknown words. The TreeTagger program assigns correct tags to 61.73% of the 'unknown' words.

Part-of-speech	Proportion of errors for unknown words	Proportion of errors for known words	Relative error for unknown words	Relative errors for known words
Verbs	3.87	0.02	33.87	0.17
Nouns	2.35	0.08	8	0.27
Adjectives	0.69	0.04	6.34	0.37
Pronouns	-	0.14	-	2.05
Numerals	0.04	0.06	1.34	2.01
Residuals	0.2	-	22.22	-

Table 9: Error rate for known and unknown words.

As expected, the error rate for known words is much lower compared to the error rates for unknown words. Similarly, the error rate is much lower for known nouns and adjectives. The evaluation of individual categories reveals that the most difficult category is the category of verb, followed by nominals, which includes nouns and adjectives, pronouns, and numerals.

In addition to the part-of-speech tags, the accuracy of the tagger varies for each genre.

Part-of-speech	Training set texts	Legal texts	Fiction texts
Verbs	10.94%	9.22%	14.14%
Nouns	35.46%	33.79%	26.61%
Pronouns	12.47%	6.68%	9.38%
Adjectives	10.24%	14.99%	9.44%

Table 10: The tagger Performance for POS in different genres.

The legal test set was compared to the training set. This revealed the similarities in style and structure of the language used.

POS	Proportion of errors in different genres				
	Legal	News	Academic	Informal	Fiction
Verbs	15.38%	26.19%	7.4%	83.13%	67.78%
Nouns	51.28%	59.52%	57.4%	8.43%	20.13%
Adjectives	33.3%	7.14%	12.9%	7.22%	8.05%
Pronouns	-	-	-	1.2%	4.02%
Numerals	-	59.52%	5.5%	-	-

Residuals	-	2.38%	16.6%	-	-

Table 11: Error rate according to each genre

Compared to the training set and legal test set, the fiction test set has a higher frequency of verbs. Most tagging errors in fiction and informal test sets are incorrectly assigned tags for verbs. This explains the low accuracy in these genres compared to other genres such as legal or news.

Type of errors in verbs	Proportion of all verb errors
Incorrect POS tag, e.g. verbs tagged as nouns or adjectives etc.	61.57%
Incorrect person/number agreement and tense	10.52%
Incorrect tense	21.05%
Incorrect person/number agreement	6.84%

Table 12: Type of errors in verbs

Thus, the most common error is verbs not being recognised as verbs. This is due to the ambiguous endings in verbs, which are the same for nominal categories. This is because coinciding word endings between verbs and nominal categories are very problematic for the TreeTagger to disambiguate (see §4.2).

Verb ending and its usage	Nominal ending	Error examples
-bis Present tense	*-bis* Genitive, singular or Plural	*dar**bis*** ‘S/he runs’ *de**bis*** ‘of sisters’
-ebs Future tense	*-ebs* Dative-accusative, singular/plural	*inan**ebs*** ‘S/he will regret this’ *saxl**ebs*** ‘to houses’

Table 13: Examples of ambiguous endings in verbs and nominals.

The other types of errors are also related to the ambiguous endings. For example, incorrectly tagged tenses (21.05%), incorrect person and number and tense (10.52%) and incorrectly tagged person and number agreement (6.84%) are due to the ambiguous endings. The word endings for tenses in Georgian are not consistent, for example, the verb ending in the *-it* can be found in plural verbs in aorist, present, aorist subjunctive, future, or imperfect tenses. On the other hand, the *-it* is the instrumental case marker for nominals. As for the markers for person of argument agreement, they are prefixal (for 1st and 2nd). However, like suffixes, prefixes are also ambiguous with nominals.

This is one of the major issues revealed in training the TreeTagger for Georgian. The TreeTagger approach is based on short beginning and end strings. For a language like Georgian with rich and complex morphology, this approach is much less likely to be useful.

To sum up, the performance of the TreeTagger program on Georgian text is 92.41%, which is at the low end of the usual TreeTagger range. This was expected given the specific problems and issues I have outlined in the paper.

6. Conclusion

In the paper, I have demonstrated the difficulties involved in retraining the TreeTagger program for Georgian. I described some general procedures such as designing a training corpus, creating a tagging (fullform) lexicon, and training the TreeTagger on Georgian texts.

In POS tagging, I used a new KATAG tagset, and rule-based enclitic tokenization approach. The KATAG tagset is a fully hierarchical-decomposable tagset, which encodes the main parts-of-speech and major word class, sub-classes of the major part-of-speech categories in Georgian (Daraselia and Hardie, fc).

I have identified some major issues in training the TreeTagger for Georgian relating to the ambiguous tags in the fullform lexicon and ambiguous wordform endings in Georgian. For ambiguous tags in the fullform lexicon, I have implemented a workaround that avoided the problem. However, no workaround was possible for ambiguous wordform endings in the context of the TreeTagger.

The performance of the TreeTagger program on Georgian text is 92.41%, which is at the low end of the usual TreeTagger range. This was expected given the specific problems and issues outlined in the paper.

For the specific issues and problems discussed in the paper, the TreeTagger is the suboptimal choice for Georgian, and other more sophisticated techniques, such as rule-based approaches are likely to do better. However, training the TreeTagger on Georgian texts is a very worthwhile goal for two main reasons. First, it allows corpus search and statistics to be applied to single token-level grammatical categories and second, and more importantly it is built into many widely used corpus query systems, such as #Lancsbox, CQPWeb, and TagAnt. The TreeTageer trained on Georgian texts means that it is accessible in these environments.

The POS tagging resources including the KATAG tagset, training corpus, tagging (fullform) lexicon, and parameter files are made available on GitHub. In addition to this, the trained TreeTagger for Georgian is now supported and available in #LancsBox (Brezina, et al., 2018) which allows search/analysis according to type, POS, or lemma.

There are a number of possible ways to improve the results in the future. KATAG is a software-agnostic tagset which means that it can be adapted and used with other techniques such as a rule-based approach. Also, obtaining more training data is likely to increase the performance. For example, adding 3 or 4 tokens of context made no difference and this could be due to data sparsity, as with 100,000 tokens it is difficult to estimate trigram and quatrogram probabilities.

References

Brezina, V., Timperley, M., & McEnery, T. (2018). #LancsBox v. 4.x [software]. Available at: http://corpora.lancs.ac.uk/lancsbox.

Charniak, E., Hendrickson, C., Jacobson, N. and Perkowitz, M. 1993. 'Equations for part of speech tagging'. In: *Proceedings of the Eleventh National Conference on Artificial Intelligence*. Menlo Park: AAAI Press/MIT Press.

Cloeren, J. 1999. Tagsets. In: van Halteren, H. ed. *Syntactic wordclass tagging*. Dordrecht: Kluwer Academic Publishers, pp. 37-54.

Daraselia, S. & Hardie, A. Forthcoming. The KATAG morphosyntactic annotation schema for Georgian.

Daraselia, S. 2019. *Computational Analysis of Morphosyntactic Categories in Georgian*. PhD thesis, University of Leeds.

Daraselia S., Sharoff, S. 2015. 'The Main Steps of the Georgian Web-Corpus Construction'. Tbilisi. Arnold Chikobava Institute of Linguistics, *Journal of Linguistics*, Volume XXXVIII, pp 52-62.

Eklund, R. 1993. 'A probabilistic tagging module based on surface pattern matching'. In: Eklund, R (ed.) (1993) NODALIDA '93 - *Proceedings of 9:e Nordiska Datalingvistikdagarna*. Stockholm: Stockholm University.

Garside, R., Leech, G. and McEnery A. 1997. *Corpus annotation: linguistic information from computer text corpora*. London: Longman.

Hardie, A. 2004. *The computational analysis of morphosyntactic categories in Urdu*. Ph.D. thesis, Lancaster University.

Hardie, A., Daraselia, S. Forthcoming. A theory for words in Georgian: traditional constructs versus corpus annotation.

Kaplan, R. & Bresnan, J. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Bresnan, J. (ed.) *The Mental Representation of Grammatical Relations*. MIT Press, pp.173-281.

Lobzhanidze, I. 2013. Morphological Analyzer and Generator of Modern Georgian Language. *In Proceedings of Georgian Language and Modern Technologies IV*, Tbilisi, pp. 82-83.

Lobzhanidze, I. 2022. *Finite-State Computational Morphology: An Analyzer and Generator for Georgian*. Springer. https://doi.org/10.1007/978-3-030-90248-3

Meurer, P. 2007. A Computational Grammar for Georgian. In Bosch, P., Gabelaia, D. & Lang, J. (eds) *Logic, Language, and Computation. TbiLLC 2007*. Springer, pp. 1-15. https://doi.org/10.1007/978-3-642-00665-4_1

Schmid, H. 1994. 'Probabilistic Part-of-Speech Tagging Using Decision Trees'. *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK.

Schmid, H. 1995. 'Improvements in Part-of-Speech Tagging with an Application to German'. *Proceedings of the ACL SIGDAT-Workshop*. Dublin, Ireland.

van Halteren, H. ed. 1999. *Syntactic wordclass tagging*. Dordrecht: Kluwer Academic Publishers.

Voutilainen, A. 1999. A short history of tagging. In: van Halteren, H. ed. *Syntactic wordclass tagging*. Dordrecht: Kluwer Academic Publishers, pp.3-4.