Submission Deadline: August 15, 2018 (23:59 CEST)

Morphological Inflection

In this project, you will implement a system that inflects words given their dictionary form and a description of the inflection. In a follow-up task, you should reconstruct the inflection description based on the lemma + inflected form. We will assign one language to each group that should be used in the tasks for language-specific feature selection.

Introduction and Guidelines

A word's form reflects syntactic and semantic features that are expressed by the word. For example, each English count noun has both singular and plural forms (apple/apples, robot/robots, process/processes), known as the inflected forms of the noun. NLP systems must be able to analyze and generate all of these inflected forms. Fortunately, inflected forms tend to be systematically related to one another.

For English, suffixes and prefixes are used to express the inflection and thus additional grammatical meaning. The rules that languages use to combine words with pre- and suffixes are referred to as a language's morphology. Languages which make extensive use of these inflections to build words are said to be morphologically-rich.

In a language with rich morphology, it is less likely that a speaker will have encountered any given word before because in these languages, new words are frequently created through suffixation, prefixation, etc. In computational approaches that rely on having encountered a word before in training data, languages with rich morphology are especially challenging since the likelihood of encountering any given word is lower. Although data sparsity is a problem for machines, it is not a problem for humans. Humans' ability to deal with rich morphology arises from two sources: The sensitivity to sub-word structure, and a knowledge that arises from this sensitivity about which morphemes exist and how they can be recombined to form new words. You will implement a system for exactly this purpose: Given an unknown word and the target inflection details, generate the inflected form.

Lemmas and Inflections

A lemma is the dictionary or canonical form of a word.[1] For example, the lemma of "speaking" and "spoke" is *speak* and the lemma of "apples" is *apple*. For English, noun lemmas are the singular form, while verb lemmas are the infinitive form. This might differ in other languages. In this exercise, your system should be able to inflect both, verbs and nouns. Some languages may require to additionally inflect adjectives.

[1] `https://en.wikipedia.org/wiki/Lemma_(morphology)`

In the following, the notation of inflection descriptions is introduced by the UniMorph Schema. The UniMorph schema consists of 23 dimensions of meaning and over 212 features. The dimensions of meaning are morphological categories, such as person, number, tense and aspect. Each represents a coherent semantic space within inflectional morphology. They include: Aktionsart, animacy, aspect, case, comparison, definiteness, deixis, evidentiality, finiteness, gender, information structure, interrogativity, mood, number, part of speech, person, polarity, politeness, switch-reference, tense, valency, and voice. These dimensions contain varying numbers of features, from just 2 for finiteness to 39 for case. Features represent the finest-grained distinctions in meaning that are possible within a given dimension [2]. A list of the dimensions and all labels is attached in the appendix. A detailed description can be found in [1]. For solving this project you don't have to know the meaning of every dimension.

Each inflected word in any given language is represented by its lemma and a bundle of UniMorph features. In general, only a few dimensions are used for the inflection of words, depending on the language. Thus, there may be dimensions that are never used in a language.

Lemma	*Inflection*	*Inflected form*	*Translation*
hablar	V;FIN;IND;PFV;PST;1;SG	hablé	I spoke
hablar	V;FIN;IND;PFV;PSF;2;SG;INFM	hablaste	you spoke
hablar	V;FIN;IND;PFV;PST;SG	habló	he/she spoke

Table 1: Three example rules for Spanish with lemmas, inflections and inflected forms

In the table above you can find examples of the UniMorph Schema. The Spanish word hablaste can be represented as the lemma hablar (sp, engl: speaking/talking) plus the inflection bundle [V;FIN;IND;PFV;PST;2;SG;INFM]. The abbreviations are as follows:

Dimension	*Label*	*Feature*
Part of Speech	V	Verb
Finiteness	FIN	Finite
Mood	IND	Indicative
Aspect	PFV	Perfective
Tense	PST	Past
Person	2	Second person
Number	SG	Singular
Politeness	INFM	Informal

Table 2: Inflection descriptions of *hablaste* (sp, engl: you spoke)

By learning the rules from Table 1, the system might have learned to stem *hablar* to *habl* by removing "ar" and append "aste" to create the second person past form. Given the lemma *dancar* (sp, engl: dancing) and the same bundle of morphological features [V;FIN;IND;PFV;PST;2;SG;INFM], this system may output: "dancaste". The correct inflected Spanish form is danzaste (sp, engl: you danced).

Input Format

The training and test files are given as text files following the same format. Each line contains one instance consisting of a lemma, the inflected target form and inflection descriptions given by

[2] `https://unimorph.github.io/schema/`

UniMorph labels separated by semicolons. The 3 elements are separated with one or multiple whitespace characters (space or tab `\t`) and each line ends with a newline character `\n`.

```
lemma target_form inflections
```

For example:

```
hablar hablaste V;FIN;IND;PFV;PSF;2;SG;INFM\n
```

The test files can be given in 2- or 3-column format. In 2-column format, the files omit the `target_form` field (Tasks 1,2) or the `inflections` field (Task 3). These fields are included in 3-column format but should be ignored by your system during testing, i.e. you are not allowed to copy or use the given target form (Tasks 1,2) or inflection description (Task 3) in any way. Your system has to accept both input formats as valid inputs. If test files in 3-column format are given, you may use the `target_form` (Tasks 1,2) or `inflections` (Task 3) to check correctness of your own results and compute the system's accuracy.

Output Format of Accuracy

```
trained on: train.txt
- training instances: 10120
tested on: test.txt
- testing instances: 640
- correct instances: 280
- accuracy: 43.750
```

Limit the floating point precision of accuracy to 3 decimal places and use the yy.xxx format.

Output Format of Group Information

```
Group L00: English
Harshita Jhavar, 2550234
Lukas Lange, 2550123
Marimuthu Kalimuthu, 2560345
```

Command Line Parameters

Your system has to accept and process the following command line parameters:

- `-g` Print your group informations and exit. Other parameters are ignored
- `-tr path` Path to the trainings file.
- `-te path` Path to the test file.
- `-a` Your system prints the accuracy following the format given before. This is only called with 3-column test files.

`-l` Your system prints each generated target form (Tasks 1,2) or inflection feature bundle (Task 3) to the standard output with one instance per line. The ordering of your output forms has to be according to the lemmas in the test file, e.g. the results for the first lemma should be output first. Your system may process the test file internally in any ordering (normal, reverse, random, concurrent, ...), but the output has to follow the original ordering.

At least one parameter `-l` or `-a` and both files (train/test) have to be specified. The path arguments can be given as absolute or relative paths and arbitrary orderings of arguments are valid. If `-l` and `-a` are given, print the accuracy after your list of results with one empty line in between. You are not allowed to print any other information. If your system is called with an invalid combination of parameters, you may exit. We will not test your system with invalid parameter combinations. Analogously, we will call your system with valid training and test files only. Though, you are free to implement an error detecting parser. In regular case, your system should be exited with Exit-Code 0 if no errors occurred during execution, otherwise use a non-zero code.

Examples for valid parameters:

```
python task1.py -g
python task1.py -l -a -tr data/train.txt -te ../test.txt
python task1.py -a -tr /home/snlp/train.txt -te /home/snlp/test.txt
python task2.py -tr /home/snlp/train.txt -te /home/snlp/test.txt -a
```

Coding Guidelines

- Prepare the files for the corresponding task `task1.py`, `task2.py` and `task3.py`. You are allowed to create additional files.
- Your code has to run with Python 3.5 or later versions.
- You can use the following python libraries:
  - Python Standard Library `https://docs.python.org/3/library/`
  - Everything from the nltk modules: nltk.stem, nltk.tokenize, nltk.corpus.stopwords
- You may use other libraries, but it's strictly forbidden to use a library that will solve this task for you. If in doubt, ask the tutor team.
- Provide detailed comments in your code, otherwise you would end up losing points.

CoNLL–SIGMORPHON 2018 Shared Task

Almost at the same time as this project, **SIGMORPHON** and **CoNLL** are hosting a shared task on universal morphological inflection [3]. Their submission deadline is July 20. You may want to take a look at this if your system performs well before their deadline. Submitting the best system for task 1 in the shared task can replace this project and will be rewarded with full points.

[3] `https://sigmorphon.github.io/sharedtasks/2018/`

Task 1: English Inflection using Minimal System (30 Points)

Implement an inflection system. Your system should compute the inflected word form given a lemma + inflection descriptions. For example, given "schielen" + "V.PTCP;PST" your system should output "geschielt". You have to report your accuracy results on the English dev set using the English training files.

(For all tasks: You have 2 training files, low and medium resource data that are varying in the number of training instances. You have to report the accuracy $((\#correct/\#total) * 100)$ *for both training files. Your systems may rely on external data like a list of stopwords.)*

You will have to implement a simple algorithm that has the following steps:

- Step 1: System should align the input and output training examples using Levenshtein distance. [4] For example:

```
schielen        geschielt       V.PTCP;PST
```

  should be aligned as

```
__schielen
geschielt_
```

- Step 2: The system now assumes that each input-output pair can be broken up into a prefixation part, a stem part, and a suffixation part, based on where the inputs or outputs have initial or trailing zeroes. You can use the list of prefixes and suffixes from the NLTK library in order to determine the rules as described in Step 3.

```
Pr   St    Su
__|schiele|n
ge|schielt|_
```

- Step 3: After this, the system extracts a set of prefix-changing rules based on the Pr pairings, and a set of suffix-changing rules based on St+Su pairings. In this example, the following suffix-changing rules are extracted:

```
n$ > $
en$ > t$
len$ > lt$
elen$ > elt$
ielen$ > ielt$
hielen$ > hielt$
chielen$ > chielt$
schielen$ > schielt$
```

  Likewise, the only prefix-changing rule extracted is the following:

```
$ > $ge
```

[4] https://en.wikipedia.org/wiki/Levenshtein_distance

For languages with few prefix changes like English, the only prefix rule will often be >, i.e. no change.
Note: All these rules are associated with the complete inflection features of an example, in this case V.PTCP;PST.

- Step 4: At generation time, the longest suffix rule that applies to a lemma form to be inflected is used. For example, if we're asked to inflect kaufen into V.PTCP;PST, we may find that, for example, en> *t* is the longest-matching suffix rule among all rules for V.PTCP;PST, which transforms:

```
$kaufen$ > $kauft$
```

  and, following this, we find the most frequently seen prefix-changing rule for the target inflection features in question that is applicable, in this case >ge and apply that:

```
$kaufen$ > $gekauft$
```

  Note: If no rule has been associated with a particular inflection feature combination, the lemma form is simply repeated. There is a heuristic to decide if a language is largely prefixing or largely suffixing. This is done by simply counting how often there is a prefix-change vs. suffix-change in going from the lemma form to the inflected form. Whenever a language is found to be largely prefixing, the system works with reversed strings throughout.

Task 2: Optimizing with language specific features (30 Points)

In this task, you will optimize the baseline model from the first task to perform better on the language assigned to your group. You have to implement further features and other functions that will increase the accuracy of your baseline system. For this, you might want to research abut the properties of your assigned language to build language-specific features. You have to describe every feature you implemented in the report. The optimizations can be language-depending features that only work with specific languages or it can be language-independent that increase accuracy for (almost) all languages. You have to report the accuracy on *dev* and *train* files of your language. Your grading will be based on the performance of your system on test data, the number and quality of features and their description in the report.

Task 3: Retrieving Inflection Features (40 Points)

In this task, you should implement a system that retrieves the inflection features from an inflected form, i.e. given a lemma + inflected form return the set of features that would have yielded the inflected form. For example, given "schielen" + "geschielt" your system should output "V.PTCP;PST". Your system should be language-dependent and achieve good results on the language assigned to your group.

A very simple baseline algorithm for this task could be as follows:

```
given lemma and target form
for every possible feature combination found during training:
  # do this using system from task 1
  compute inflected form using lemma + feature combination
  if computed form equals given target form:
    return feature combination
return {}  # empty set
```

Note: You don't have to use this algorithm in your implementation and we strongly recommend that you use a different solution. You can solve this task by using any algorithm you like and you can make use of any features you can derive from the training data.

In the report, you have to describe the basic idea of your algorithm and give descriptions for all the features/optimizations you implemented. You can find example of descriptions in the report template Section 2. Your descriptions should be at least as detailed as the examples. Report the accuracy for your assigned language and the increases by adding more complex features. Your grading will be based on the performance of your system on test data, the number and quality of features and their description in the report.

Submission Instructions

- Submit only 1 archive file in the ZIP format with name containing the LanguageID and MN of all the team members, e.g.:

  Project_LXX_MatriculationNumber1_MatriculationNumber2.zip

- Provide in the archive: i. all your code, accompanied with sufficient comments
  ii. the PDF report with answers, solutions, plots and brief instructions on executing your code
  iii. a README file with the group member names, matriculation numbers and emails
  iv. Data necessary to reproduce your results
  v. the following files must be contained in your archive: report.pdf, task1.py (for English), task2.py and task3.py (for your Language)
- The subject of your submission mail **must** contain the string "[SNLP]" (including the braces) and explicitly denoting that it is an exercise submission, e.g:

  [SNLP] Project Submission LXX MatriculationNumber1_MatriculationNumber2

- Submit to `snlp_tutors_2018@lsv.uni-saarland.de`
- Note that if you submit to any other email than the above, it will not be graded at any cost.
- Late submissions will not be accepted. (i.e. Late submissions would automatically get 0 points for the whole project.)
- You have to follow the instructions from the coding guidelines and the guidelines mentioned in the report template. Otherwise we will subtract points.

- **Copying solutions is strictly forbidden**. If any form of plagiarism is found out, then all teams involved in the practice would get **0 points** and they will be **excluded from the course**. Note that copying from web resources also counts as **plagiarism**, unless it is properly cited.

References

[1] Sylak-Glassman, J. (2016). The Composition and Use of the Universal Morphological Feature Schema (UniMorph Schema). Department of Computer Science, Johns Hopkins University. Link `https://unimorph.github.io/doc/unimorph-schema.pdf`

[2] Kibrik, Aleksandr E. (1998). "Archi". In Andrew Spencer; Arnold M. Zwicky. The Handbook of Morphology. Blackwell Publishers. pp. 455-476.

Appendix

Taken from `https://unimorph.github.io/doc/unimorph-schema.pdf`[1] pages 60-71

7 Appendix 1: Full Alphabetical Listing of Dimensions and Features

The following table lists the dimensions of meaning, sorted alphabetically, along with their features, alphabetized on the feature labels.

Dimension	*Feature*	*Label*
Aktionsart	Accomplishment	ACCMP
Aktionsart	Achievement	ACH
Aktionsart	Activity	ACTY
Aktionsart	Atelic	ATEL
Aktionsart	Durative	DUR
Aktionsart	Dynamic	DYN
Aktionsart	Punctual	PCT
Aktionsart	Semelfactive	SEMEL
Aktionsart	Stative	STAT
Aktionsart	Telic	TEL
Animacy	Animate	ANIM
Animacy	Human	HUM
Animacy	Inanimate	INAN
Animacy	Non-human	NHUM
Argument Marking	3.SG Object (from feature template)	ARGAC3S
Aspect	Habitual	HAB
Aspect	Imperfective	IPFV
Aspect	Iterative	ITER
Aspect	Perfective	PFV
Aspect	Perfect	PRF
Aspect	Progressive	PROG
Aspect	Prospective	PROSP
Case	Ablative	ABL
Case	Absolutive	ABS
Case	Accusative	ACC
Case	Allative	ALL
Case	Near, in front of	ANTE
Case	Approximative	APPRX
Case	Next to	APUD
Case	At	AT
Case	Aversive	AVR
Case	Benefactive	BEN
Case	Essive-modal	BYWAY
Case	Near	CIRC
Case	Comitative	COM
Case	Comparative	COMPV
Case	Dative	DAT
Case	Equative	EQTV
Case	Ergative	ERG
Case	Essive	ESS
Case	Formal	FRML

Case	Genitive	GEN
Case	In	IN
Case	Instrumental	INS
Case	Among	INTER
Case	Nominative	NOM
Case	Nominative, S-only	NOMS
Case	On	ON
Case	On (horizontal)	ONHR
Case	On (vertical)	ONVR
Case	Behind	POST
Case	Privative	PRIV
Case	Prolative/translative	PROL
Case	Proprietive	PROPR
Case	Proximate	PROX
Case	Purposive	PRP
Case	Partitive	PRT
Case	Relative	REL
Case	Distal	REM
Case	Under	SUB
Case	Terminative	TERM
Case	Translative	TRANS
Case	Versative	VERS
Case	Vocative	VOC
Comparison	Absolute	AB
Comparison	Comparative	CMPR
Comparison	Equative	EQT
Comparison	Relative	RL
Comparison	Superlative	SPRL
Definiteness	Definite	DEF
Definiteness	Indefinite	INDF
Definiteness	Non-Specific	NSPEC
Definiteness	Specific	SPEC
Deixis	Above	ABV
Deixis	Below	BEL
Deixis	Even	EVEN
Deixis	Medial	MED
Deixis	No Reference Point, Distal	NOREF
Deixis	Invisible	NVIS
Deixis	Phoric, situated in discourse	PHOR
Deixis	Proximate	PROX
Deixis	First Person Reference Point	REF1
Deixis	Second Person Reference Point	REF2
Deixis	Remote	REMT
Deixis	Visible	VIS
Evidentiality	Assumed	ASSUM
Evidentiality	Auditory	AUD
Evidentiality	Direct	DRCT

Evidentiality	Firsthand	FH
Evidentiality	Hearsay	HRSY
Evidentiality	Inferred	INFER
Evidentiality	Non-firsthand	NFH
Evidentiality	Non-visual sensory	NVSEN
Evidentiality	Quotative	QUOT
Evidentiality	Reported	RPRT
Evidentiality	Sensory	SEN
Finiteness	Finite	FIN
Finiteness	Nonfinite	NFIN
Gender	Bantu Noun Classes	BANTU1-23
Gender	Feminine	FEM
Gender	Masculine	MASC
Gender	Nakh-Daghestanian Noun Classes	NAKH1-8
Gender	Neuter	NEUT
Information Structure	Focus	FOC
Information Structure	Topic	TOP
Interrogativity	Declarative	DECL
Interrogativity	Interrogative	INT
Language-Specific Features	varies by language	LGSPEC1
Language-Specific Features	varies by language	LGSPEC2
Mood	Admirative	ADM
Mood	Australian Non-Purposive	AUNPRP
Mood	Australian Purposive	AUPRP
Mood	Conditional	COND
Mood	Debitive	DEB
Mood	Deductive	DED
Mood	Imperative-Jussive	IMP
Mood	Indicative	IND
Mood	Intentive	INTEN
Mood	Irrealis	IRR
Mood	Likely	LKLY
Mood	Obligative	OBLIG
Mood	Optative-Desiderative	OPT
Mood	Permissive	PERM
Mood	Potential	POT
Mood	General Purposive	PURP
Mood	Realis	REAL
Mood	Subjunctive	SBJV
Mood	Simulative	SIM
Number	Dual	DU
Number	Greater paucal	GPAUC
Number	Greater plural	GRPL
Number	Inverse	INVN
Number	Paucal	PAUC
Number	Plural	PL
Number	Singular	SG

Number	Trial	TRI
Part of Speech	Adjective	ADJ
Part of Speech	Adposition	ADP
Part of Speech	Adverb	ADV
Part of Speech	Article	ART
Part of Speech	Auxiliary	AUX
Part of Speech	Classifier	CLF
Part of Speech	Complementizer	COMP
Part of Speech	Conjunction	CONJ
Part of Speech	Determiner	DET
Part of Speech	Interjection	INTJ
Part of Speech	Noun	N
Part of Speech	Numeral	NUM
Part of Speech	Particle	PART
Part of Speech	Pronoun	PRO
Part of Speech	Proper Name	PROPN
Part of Speech	Verb	V
Part of Speech	Converb	V.CVB
Part of Speech	Masdar	V.MSDR
Part of Speech	Participle	V.PTCP
Person	Zero person	0
Person	First person	1
Person	Second person	2
Person	Third person	3
Person	Fourth person	4
Person	Exclusive	EXCL
Person	Inclusive	INCL
Person	Obviative	OBV
Person	Proximate	PRX
Polarity	Positive	POS
Polarity	Negative	NEG
Politeness	Avoidance style	AVOID
Politeness	Colloquial	COL
Politeness	Formal, Referent Elevating	ELEV
Politeness	Formal register	FOREG
Politeness	Formal	FORM
Politeness	High status	HIGH
Politeness	Formal, Speaker Humbling	HUMB
Politeness	Informal	INFM
Politeness	Literary	LIT
Politeness	Low status	LOW
Politeness	Polite	POL
Politeness	High status, elevated	STELEV
Politeness	High status, supreme	STSUPR
Possession	Alienable possession	ALN
Possession	Inalienable possession	NALN
Possession	Possession by 1.DU	PSS1D

Possession	Possession by 1.DU.EXCL	PSS1DE
Possession	Possession by 1.DU.INCL	PSS1DI
Possession	Possession by 1.PL	PSS1P
Possession	Possession by 1.PL.EXCL	PSS1PE
Possession	Possession by 1.PL.INCL	PSS1PI
Possession	Possession by 1.SG	PSS1S
Possession	Possession by 2.DU	PSS2D
Possession	Possession by 2.DU.FEM	PSS2DF
Possession	Possession by 2.DU.MASC	PSS2DM
Possession	Possession by 2.PL	PSS2P
Possession	Possession by 2.PL.FEM	PSS2PF
Possession	Possession by 2.PL.MASC	PSS2PM
Possession	Possession by 2.SG	PSS2S
Possession	Possession by 2.SG.FEM	PSS2SF
Possession	Possession by 2.SG.FORM	PSS2SFORM
Possession	Possession by 2.SG.INFM	PSS2SINFM
Possession	Possession by 2.SG.MASC	PSS2SM
Possession	Possession by 3.DU	PSS3D
Possession	Possession by 3.DU.FEM	PSS3DF
Possession	Possession by 3.DU.MASC	PSS3DM
Possession	Possession by 3.PL	PSS3P
Possession	Possession by 3.PL.FEM	PSS3PF
Possession	Possession by 3.PL.MASC	PSS3PM
Possession	Possession by 3.SG	PSS3S
Possession	Possession by 3.SG.FEM	PSS3SF
Possession	Possession by 3.SG.MASC	PSS3SM
Possession	Possessed	PSSD
Switch-Reference	SR among NPs in any argument position	CN_R_MN
Switch-Reference	DS	DS
Switch-Reference	DS Adverbial	DSADV
Switch-Reference	Logophoric	LOG
Switch-Reference	Open Reference	OR
Switch-Reference	Sequential Multiclausal Aspect	SEQMA
Switch-Reference	Simultaneous Multiclausal Aspect	SIMMA
Switch-Reference	SS	SS
Switch-Reference	SS Adverbial	SSADV
Tense	Within 1 day	1DAY
Tense	Future	FUT
Tense	Hodiernal	HOD
Tense	Immediate	IMMED
Tense	Present	PRS
Tense	Past	PST
Tense	Recent	RCT
Tense	Remote	RMT
Valency	Applicative	APPL
Valency	Causative	CAUS
Valency	Ditransitive	DITR

Valency	Impersonal	IMPRS
Valency	Intransitive	INTR
Valency	Reciprocal	RECP
Valency	Reflexive	REFL
Valency	Transitive	TR
Voice	Accompanier Focus	ACFOC
Voice	Active	ACT
Voice	Agent Focus	AGFOC
Voice	Antipassive	ANTIP
Voice	Beneficiary Focus	BFOC
Voice	Conveyed Focus	CFOC
Voice	Direct	DIR
Voice	Instrument Focus	IFOC
Voice	Inverse	INV
Voice	Location Focus	LFOC
Voice	Middle	MID
Voice	Passive	PASS
Voice	Patient Focus	PFOC

Table 35: Dimensions of meaning presented alphabetically with their features sorted alphabetically by feature label

8 Appendix 2: Full Alphabetical Listing of Features and Dimensions

This appendix contains feature labels sorted alphabetically, with a short gloss of the feature and the dimension in which it belongs.

Label	*Feature*	*Dimension*
0	Zero person	Person
1	First person	Person
2	Second person	Person
3	Third person	Person
4	Fourth person	Person
1DAY	Within 1 day	Tense
AB	Absolute	Comparison
ABL	Ablative	Case
ABS	Absolutive	Case
ABV	Above	Deixis
ACC	Accusative	Case
ACCMP	Accomplishment	Aktionsart
ACFOC	Accompanier Focus	Voice
ACH	Achievement	Aktionsart
ACT	Active	Voice
ACTY	Activity	Aktionsart
ADJ	Adjective	Part of Speech
ADM	Admirative	Mood
ADP	Adposition	Part of Speech
ADV	Adverb	Part of Speech
AGFOC	Agent Focus	Voice
ALL	Allative	Case
ALN	Alienable Possession	Possession
ANIM	Animate	Animacy
ANTE	Near, in front of	Case
ANTIP	Antipassive	Voice
APPL	Applicative	Valency
APPRX	Approximative	Case
APUD	Next to	Case
ARGAC3S	3.SG Object (from feature template)	Argument Marking
ART	Article	Part of Speech
ASSUM	Assumed	Evidentiality
AT	At	Case
ATEL	Atelic	Aktionsart
AUD	Auditory	Evidentiality
AUNPRP	Australian Non-Purposive	Mood
AUPRP	Australian Purposive	Mood
AUX	Auxiliary	Part of Speech
AVOID	Avoidance style	Politeness
AVR	Aversive	Case
BANTU1-23	Bantu Noun Classes	Gender

BEL	Below	Deixis
BEN	Benefactive	Case
BFOC	Beneficiary Focus	Voice
BYWAY	Essive-modal	Case
CAUS	Causative	Valency
CFOC	Conveyed Focus	Voice
CIRC	Near	Case
CLF	Classifier	Part of Speech
CMPR	Comparative	Comparison
CN_R_MN	SR among NPs in any argument position	Switch-Reference
COL	Colloquial	Politeness
COM	Comitative	Case
COMP	Complementizer	Part of Speech
COMPV	Comparative	Case
COND	Conditional	Mood
CONJ	Conjunction	Part of Speech
DAT	Dative	Case
DEB	Debitive	Mood
DECL	Declarative	Interrogativity
DED	Deductive	Mood
DEF	Definite	Definiteness
DET	Determiner	Part of Speech
DIR	Direct	Voice
DITR	Ditransitive	Valency
DRCT	Direct	Evidentiality
DS	DS	Switch-Reference
DSADV	DS Adverbial	Switch-Reference
DU	Dual	Number
DUR	Durative	Aktionsart
DYN	Dynamic	Aktionsart
ELEV	Formal, Referent Elevating	Politeness
EQT	Equative	Comparison
EQTV	Equative	Case
ERG	Ergative	Case
ESS	Essive	Case
EVEN	Even	Deixis
EXCL	Exclusive	Person
FEM	Feminine	Gender
FH	Firsthand	Evidentiality
FIN	Finite	Finiteness
FOC	Focus	Information Structure
FOREG	Formal register	Politeness
FORM	Formal	Politeness
FRML	Formal	Case
FUT	Future	Tense
GEN	Genitive	Case
GPAUC	Greater paucal	Number

GRPL	Greater plural	Number
HAB	Habitual	Aspect
HIGH	High status	Politeness
HOD	Hodiernal	Tense
HRSY	Hearsay	Evidentiality
HUM	Human	Animacy
HUMB	Formal, Speaker Humbling	Politeness
IFOC	Instrument Focus	Voice
IMMED	Immediate	Tense
IMP	Imperative-Jussive	Mood
IMPRS	Impersonal	Valency
IN	In	Case
INAN	Inanimate	Animacy
INCL	Inclusive	Person
IND	Indicative	Mood
INDF	Indefinite	Definiteness
INFER	Inferred	Evidentiality
INFM	Informal	Politeness
INS	Instrumental	Case
INT	Interrogative	Interrogativity
INTEN	Intentive	Mood
INTER	Among	Case
INTJ	Interjection	Part of Speech
INTR	Intransitive	Valency
INV	Inverse	Voice
INVN	Inverse	Number
IPFV	Imperfective	Aspect
IRR	Irrealis	Mood
ITER	Iterative	Aspect
LFOC	Location Focus	Voice
LGSPEC1	Varies by language	Language-Specific Features
LGSPEC2	Varies by language	Language-Specific Features
LIT	Literary	Politeness
LKLY	Likely	Mood
LOG	Logophoric	Switch-Reference
LOW	Low status	Politeness
MASC	Masculine	Gender
MED	Medial	Deixis
MID	Middle	Voice
N	Noun	Part of Speech
NAKH1-8	Nakh-Daghestanian Noun Classes	Gender
NALN	Inalienable Possession	Possession
NEG	Negative	Polarity
NEUT	Neuter	Gender
NFH	Non-firsthand	Evidentiality
NFIN	Nonfinite	Finiteness
NHUM	Non-human	Animacy

NOM	Nominative	Case
NOMS	Nominative, S-only	Case
NOREF	No Reference Point, Distal	Deixis
NSPEC	Non-Specific	Definiteness
NUM	Numeral	Part of Speech
NVIS	Invisible	Deixis
NVSEN	Non-visual sensory	Evidentiality
OBLIG	Obligative	Mood
OBV	Obviative	Person
ON	On	Case
ONHR	On (horizontal)	Case
ONVR	On (vertical)	Case
OPT	Optative-Desiderative	Mood
OR	Open Reference	Switch-Reference
PART	Particle	Part of Speech
PASS	Passive	Voice
PAUC	Paucal	Number
PCT	Punctual	Aktionsart
PERM	Permissive	Mood
PFOC	Patient Focus	Voice
PFV	Perfective	Aspect
PHOR	Phoric, situated in discourse	Deixis
PL	Plural	Number
POL	Polite	Politeness
POS	Positive	Person
POS	Positive	Polarity
POST	Behind	Case
POT	Potential	Mood
PRF	Perfect	Aspect
PRIV	Privative	Case
PRO	Pronoun	Part of Speech
PROG	Progressive	Aspect
PROL	Prolative/translative	Case
PROPN	Proper Name	Part of Speech
PROPR	Proprietive	Case
PROSP	Prospective	Aspect
PROX	Proximate	Case
PROX	Proximate	Deixis
PRP	Purposive	Case
PRS	Present	Tense
PRT	Partitive	Case
PRX	Proximate	Person
PSS1D	Possession by 1.DU	Possession
PSS1DE	Possession by 1.DU.EXCL	Possession
PSS1DI	Possession by 1.DU.INCL	Possession
PSS1P	Possession by 1.PL	Possession
PSS1PE	Possession by 1.PL.EXCL	Possession

PSS1PI	Possession by 1.PL.INCL	Possession
PSS1S	Possession by 1.SG	Possession
PSS2D	Possession by 2.DU	Possession
PSS2DF	Possession by 2.DU.FEM	Possession
PSS2DM	Possession by 2.DU.MASC	Possession
PSS2P	Possession by 2.PL	Possession
PSS2PF	Possession by 2.PL.FEM	Possession
PSS2PM	Possession by 2.PL.MASC	Possession
PSS2S	Possession by 2.SG	Possession
PSS2SF	Possession by 2.SG.FEM	Possession
PSS2SFORM	Possession by 2.SG.FORM	Possession
PSS2SINFM	Possession by 2.SG.INFM	Possession
PSS2SM	Possession by 2.SG.MASC	Possession
PSS3D	Possession by 3.DU	Possession
PSS3DF	Possession by 3.DU.FEM	Possession
PSS3DM	Possession by 3.DU.MASC	Possession
PSS3P	Possession by 3.PL	Possession
PSS3PF	Possession by 3.PL.FEM	Possession
PSS3PM	Possession by 3.PL.MASC	Possession
PSS3S	Possession by 3.SG	Possession
PSS3SF	Possession by 3.SG.FEM	Possession
PSS3SM	Possession by 3.SG.MASC	Possession
PSSD	Possessed	Possession
PST	Past	Tense
PURP	General Purposive	Mood
QUOT	Quotative	Evidentiality
RCT	Recent	Tense
REAL	Realis	Mood
RECP	Reciprocal	Valency
REF1	First Person Reference Point	Deixis
REF2	Second Person Reference Point	Deixis
REFL	Reflexive	Valency
REL	Relative	Case
REM	Distal	Case
REMT	Remote	Deixis
RL	Relative	Comparison
RMT	Remote	Tense
RPRT	Reported	Evidentiality
SBJV	Subjunctive	Mood
SEMEL	Semelfactive	Aktionsart
SEN	Sensory	Evidentiality
SEQMA	Sequential Multiclausal Aspect	Switch-Reference
SG	Singular	Number
SIM	Simulative	Mood
SIMMA	Simultaneous Multiclausal Aspect	Switch-Reference
SPEC	Specific	Definiteness
SPRL	Superlative	Comparison

SS	SS	Switch-Reference
SSADV	SS Adverbial	Switch-Reference
STAT	Stative	Aktionsart
STELEV	High status, elevated	Politeness
STSUPR	High status, supreme	Politeness
SUB	Under	Case
TEL	Telic	Aktionsart
TERM	Terminative	Case
TOP	Topic	Information Structure
TR	Transitive	Valency
TRANS	Translative	Case
TRI	Trial	Number
V	Verb	Part of Speech
V.CVB	Converb	Part of Speech
V.MSDR	Masdar	Part of Speech
V.PTCP	Participle	Part of Speech
VERS	Versative	Case
VIS	Visible	Deixis
VOC	Vocative	Case

Table 36: Features sorted alphabetically by their label, with their short gloss and dimension of meaning indicated