UNIVERSITÀ DEGLI STUDI DI VERONA

DEPARTMENT OF

Filologia, Letteratura e Linguistica

GRADUATE SCHOOL OF

Scuola di Dottorato in Studi Umanistici

DOCTORAL PROGRAM IN

Linguistics

WITH THE FINANCIAL CONTRIBUTION OF

Università degli Studi di Verona

Cycle

XXVII

TITLE OF THE DOCTORAL THESIS

A computational morphological analysis of Italian verbal system

IN CO-TUTELLE DE THÈSE WITH THE UNIVERSITY OF

Université de Toulouse II - Jean Jaurès

S.S.D.

L-LIN/01

Doctoral Student:	Dott.	Matteo PASCOLI
Tutor:	Prof.ssa	Paola COTTICELLI KURRAS
Tutor:	Prof.	Fabio MONTERMINI
Coordinator:	Prof.ssa	Birgit ALBER

Index

Abstract

Verbal inflection in Italian, as it happens in other romance languages, is complex. Its complexity derives not just from the number of forms, each coupled with a distinct set of morphosyntactic properties –mood, tense, person, number– but also, especially, from the variability of said forms.

While the process of structuring the verbal lexicon into classes can account for the variability in the ending of the inflected forms (the desinence), it can not account for the variability in the stem part, because there would be too many classes needed to classify these phenomena of *allomorphism*. The traditional approach requires the speaker to memorize a list of the forms whose stem part is not identical to other forms of the same paradigm, or in particular to the presentation form of the lexeme (infinitive for Italian verbs), as *exceptions*.

In the last twenty years, there has been much interest in studying the *paradigmatic distribution of allomorphy*, or the way in which the variation (the traditional "irregularity") between forms of a paradigm (not only of verbs, but also of nouns and adjectives) rests on regular schemes.

Said interest has at least three directions. The first one is purely technical, suggested by the desire to pack morphological information as dense as possible to build computing efficient applications that parse, interpret, analyse, translate or produce texts (or speech), without the need to peruse enormous amounts of redundant data. The second one is cognitive: studies

on the analogical associations and on how these associations form patterns and schemes can contribute to the insight on how our brain works. The third one is didactical, since the learning of languages can greatly benefit from the knowledge on such patterns of association and their operation.

The practical approach of these researches has the goal of analysing the paradigmatical structure of inflection, that is, to decompose the paradigm in zones where the forms are realized on possibly distinct basic stems, and to examine the formal relations (on the phonological level) between these basic stems, studying the chains of predictability that permit us, the speakers, to handle both regular and irregular lexemes.

With this work I have carried an analysis of the Italian verbal system. Following a *Word and Paradigm* point of view, and researches who have studied the inflectional morphology with paradigmatic approach, my goal was to build algorithms and programs to calculate relations between the word forms comprising the whole flexion of a sample of Italian verbs. The set of evaluated verbs covers all models of conjugation, including highly irregular verbs.

The contribution to inflectional morphology articulates on these points:

– the analysis is on the phonetic forms, as opposed to orthographic forms. I have thus developed a database for generating forms for all paradigm cells in their phonetic transcription.

– the analysis is fully automated. I have developed all the algorithms needed in Java language, so that after a change in the database (for further lexemes, or possibly correction of mistakes), or even the switch to another set of data, for analysing other languages, the whole computation takes few minutes to run.

– the analysis does not depend on the supposition that inflection happens at the end of the word, or by suffixation: the algorithms developed can work with discontinuous flexion (as found in Semitic languages, or partially in German and Greek, for example) with the same principles.

Abstract (Italiano)

La flessione verbale in italiano, come avviene per le altre lingue romanze, è complessa. La sua complessità deriva non solo dal numero di forme, ciascuna associata a un insieme distinto di proprietà morfosintattiche –modo, tempo, persona, numero– ma anche dalla variabilità di tali forme.

Benché il procedimento di strutturare il lessico verbale in classi possa rendere conto della variabilità nella parte terminale delle forme flesse (le desinenze), esso non può rendere conto della variabilità nella parte tematica, poiché sarebbe necessario un numero troppo alto di classi per tenere conto di tutti questi fenomeni di *allomorfia*. L'approccio tradizionale richiede che il parlante memorizzi una lista delle forme la cui parte tematica differisce dalle altre forme dello stesso paradigma, o in particolare dalla forma di presentazione del lessema (per i verbi italiani, l'infinito), come *eccezioni*.

Negli ultimi vent'anni, si è mostrato un particolare interesse nello studio della distribuzione paradigmatica dell'allomorfia, ovvero, delle modalità in cui la variabilità (la tradizionale *irregolarità*) tra forme di un dato paradigma (non solo per i verbi, ma anche per i nomi e gli aggettivi) posa su schemi regolari.

Questo interesse ha almeno tre motivazioni. La prima è puramente tecnica, basata sul desiderio di organizzare l'informazione morfologica nel modo più compatto possibile, sviluppando applicazioni *software* efficienti che analizzino, interpretino, traducano o producano testi (o parlato), senza la

necessità di consultare quantitativi enormi di dati ridondanti. La seconda è nel dominio delle scienze cognitive: gli studi sulle associazioni analogiche e su come queste associazioni formino schemi regolari possono contribuire alla comprensione di come funziona il nostro cervello. La terza è sul piano didattico, poiché lo studio e l'insegnamento delle lingue possono trarre grande beneficio dalla conoscenza di tali schemi di associazione e del loro funzionamento.

L'approccio pratico di queste ricerche consiste nell'analisi della struttura paradigmatica della flessione, effettuata scomponendo il paradigma in zone che differiscono potenzialmente dalla forma del tema a partire dal quale si realizzano le singole forme flesse, ed esaminando le relazioni formali (sul livello fonologico) tra queste basi tematiche, studiando le catene di predicibilità che permettono a noi parlanti di gestire sia i lessemi regolari che quelli irregolari.

In questo lavoro ho compiuto un'analisi del sistema verbale italiano. Seguendo il punto di vista *Word and Paradigm*, e i ricercatori che si sono occupati di morfologia flessiva con un approccio paradigmatico, il mio obiettivo era sviluppare algoritmi e programmi per calcolare le relazioni tra le forme della coniugazione dei verbi italiani. L'insieme dei verbi considerati copre tutti i modelli di coniugazione, inclusi i verbi altamente irregolari.

Il contributo alla morfologia flessiva si articola nei seguenti punti:

– l'analisi è fatta sulle forme fonetiche, non sulle forme ortografiche. Per questo ho sviluppato un *database* per generare le forme di tutte le celle del paradigma nella trascrizione fonetica.

– l'analisi è completamente automatica. Ho sviluppato gli algoritmi necessari tramite il linguaggio di programmazione Java, così che ad ogni modifica del database (per aggiungere lessemi, o eventualmente applicare correzioni), o anche al passaggio di un insieme completamente diverso di dati, per analizzare altre lingue, l'intera elaborazione richiede pochi minuti di calcolo.

– l'analisi non dipende dal presupposto che la flessione avvenga nella parte terminale della parola, ovvero per suffissazione: gli algoritmi sviluppati funzionano anche per la flessione discontinua (come per esempio nelle lingue semitiche, o in parte in greco e in tedesco) con gli stessi principi.

Abstract (Français)

La flexion verbale de l'italien, comme pour les autres langues romanes, est complexe. Sa complexité dérive non seulement du nombre des formes, chacune associée à un ensemble de propriétés morphosyntaxiques –mode, temps, personne, nombre– mais aussi, en particulier, de la variabilité de ces formes.

Bien que le procès de structuration du lexique verbal en classes puisse rendre compte de la variation dans la partie terminale des formes fléchies (la désinence), il ne peut pas rendre compte de la variation dans la partie thématique, puisque un nombre excessif de classes serait nécessaire pour classifier les phénomènes d'*allomorphie*. L'approche traditionnelle demande au locuteur de mémoriser une liste de formes dont la partie thématique diffère des autres formes du même paradigme, ou en in particulier de la forme de présentation du lexème (l'infinitif pour les verbes italiens), comme *exceptions*.

Dans vingt dernières années il y a eu un grand intérêt pour l'étude de la *distribution paradigmatique de l'allomorphie*, où la façon dont la variation (l'"irrégularité" traditionnelle) entre formes d'un paradigme (pas seulement des verbes, mais aussi des noms et des adjectifs) repose sur des patrons réguliers.

Cet intérêt a pris au moins trois directions. La première est strictement technique, suggérée par le désir de compresser l'information morphologique

le plus densément possible pour développer des logiciels efficaces pour analyser, interpréter, traduire ou produire des textes (ou de la parole), sans la nécessité d'élaborer des quantités énormes de données redondantes. La deuxième se situe dans le domaine des sciences cognitives: les études sur les associations d'analogie et sur la manière dont ces associations forment des patrons et des combinaisons réguliers peuvent contribuer grandement à la connaissance du fonctionnement du cerveau. La troisième se situe sur le plan didactique, puisque l'apprentissage des langues peut bénéficier considérablement de la connaissance de tels patrons d'association et de leur exploitation.

L'approche pratique de ces recherches a pour bout d'analyser la structure paradigmatique de la flexion, c'est-à-dire, de décomposer le paradigme en zones qui diffèrent potentiellement dans la forme du thème sur lequel se réalisent les formes fléchies, et d'examiner les relations formelles (au niveau phonologique) entre ces bases thématiques, en étudiant les chaînes de prédictibilité qui permettent à nous, les locuteurs, de gérer soit les lexèmes réguliers, soit les irréguliers.

Dans ce travail j'ai réalisé une analyse du système verbal italien. Suivant le point de vue *Word and Paradigm*, et les chercheurs qui ont étudié la morphologie flexionnelle dans une approche paradigmatique, mon but était de construire des algorithmes et des logiciels pour calculer les relations entre formes verbales qui embrassent la flexion entière d'un échantillon de

verbes italiens. L'ensemble des verbes évalués couvre tous les patrons de conjugaison, y compris les verbes hautement irreguliers.

L'apport à la morphologie flexionnelle s'articule autour de ces points:

– l'analyse se fait sur les formes phonétiques, au lieu que sur les formes orthographiques. J'ai donc realisé une base de donnés pour génerer les formes pour toutes les cases du paradigme en transcription phonétique.

– l'analyse est entièrement automatisée. J'ai construit tous les algorithmes nécessaires en langage Java, de façon qu'après chaque changement dans la base de données (pour ajouter des lexèmes, ou pour corriger des erreurs), ou encore pour passer à un ensemble différente de données pour analyser d'autres langues, la computation entière prend quelques minutes.

– l'analyse ne dépend pas de la supposition que la flexion a lieu à la fin du mot, c'est-à-dire par suffixation: les algorithmes réalisés peuvent travailler avec une flexion discontinue (comme l'on trouve, par exemple, dans les langues sémitiques, ou partiellement en allemand ou en grec) selon les mêmes principes.

Introduction

Verbal inflection in Italian, as it happens in other romance languages, is complex. Its complexity derives not just from the number of forms, each coupled with a distinct set of morphosyntactic properties –mood, tense, person, number– but also, especially, from the variability of said forms.

Traditionally, each conjugated form is analysed as the linear union of a stem and an ending, or desinence, subject to further regular phonological adjustments. The variability is observed both in the ending and in the stem of verbal forms. Variability in the ending is gracefully handled by dividing the verbal lexemes into few canonical conjugation classes: from three classes *-are*, *-ere*, and *-ire*, to five classes *-are*, *-ere* (paroxytone), *-ere* (proparoxytone), *-ire* (without *-sc-* extension) and *-ire* (with *-sc-* extension), depending on the author. Ending variation depends solely on the conjugation class, whose knowledge is strictly coupled with the lexeme, except in very few cases: ESSERE ‘to be’, AVERE ‘to have’, FARE ‘to do’, and mono-syllabic forms as founds in verbs DARE ‘to give’, DIRE ‘to say’, STARE ‘to stay’.

The same process, of structuring the verbal lexicon into classes, could not be used to account for the variability in the stem part of verbs, because –to state it simply– there would be too many classes needed to classify these phenomena of *allomorphism*. The traditional approach requires the speaker to memorize a list of the forms whose stem part is not identical to other

forms of the same paradigm, or in particular to the presentation form of the lexeme (infinitive for Italian verbs), as *exceptions*.

In Latin, the variation on the ending part is also dictated by the membership of a lexeme to a conjugation class, except for a handful of verbs: EDO, EO, FERO, FIO, SUM, VOLO and their derivatives. Latin verbs have three stems: imperfective, perfective and supine. The imperfective stem is the basis for Indicative Present, Imperfect and Future, Subjunctive Present and Imperfect, Imperative, Infinitive, Present Participle, Gerund and Gerundive, and simple forms of the Passive conjugation. The perfective stem is the basis for Indicative Perfect, Pluperfect and Future Second, Subjunctive Perfect and Pluperfect. The supine stem is the basis for Perfect Participle, Supine, and of course the composite forms, which are built on the Perfect Participle.

Latin verbs show allomorphy between their three stems, due to morphonological phenomena not reconducible to the phonetic context. For example, reduplication, vocalic ablauting, nasal insertion, sibilant insertion, spirantization, etc., are phenomena that occur between stems of some verbs: *laedo / laesi / laesum* 'to hurt', *pingo / pixi / pictum* 'to paint', *rego / rēgi / rectum* 'to rule', *rumpo / rūpi / ruptum* 'to break', *tondeo / totondi / tonsum* 'to shear', etc. The important point, here, is that saying that these alternations occur between different stems of a lexeme means that they occur systematically with all forms based on each of those stems, and in Latin there is no variation triggered by person, number, mood or tense.

The proto-romance phase shattered this system. New sounds and new accentuation schemes emerging from the conjugation of the verbs created new alternation patterns throughout the paradigm. Probably at first they were regular and predictable, probably they were sporadic to the point that even a single speaker had some variation in the production. But the key point is that these variations, the allomorphy, were no more only detectable between principal parts, the imperfective conjugation, the perfective conjugation, and the supine/participial conjugation: there was variation even inside a single mood/tense combination, depending on the person and number, or inside a person/number combination, depending on mood and tense; and forms with completely different sets of morphosyntactic properties could be on the same side of an allomorphic variation.

This situation continued, regularised and further evolved, in modern Romance languages. Conjugation of so called "irregular" verbs could be naively described as a mess, and indeed studying the Italian or French verbs is more complicated than studying a Latin verb. Hovewer, it was observed (Maiden 1992; Pirrelli 2000) how alternations on verbal stems meet a surprisingly regular distribution, which is no more dictated –at least, not only– by phonological context. This regularity reflects the organisation of the verbal *paradigm*, or the ensemble of all forms, into *morphomes*, purely formal entities independent from morphosyntactical traits (Aronoff 1994). Morphomes arrange themselves in the verbal paradigm, synchronically

(because, diachronically, this arrangement is mainly based on phonological causes), on the basis of analogical cognitive mechanisms.

In the last twenty years, there has been much interest in studying the *paradigmatic distribution of allomorphy*, or the way in which the variation (the traditional "irregularity") between forms of a paradigm (not only of verbs, but also of nouns and adjectives) rests on regular schemes.

Said interest has at least three directions. The first one is purely technical, suggested by the desire to pack morphological information as dense as possible to build computing efficient applications that parse, interpret, analyse, translate or produce texts (or speech), without the need to peruse enormous amounts of redundant data. The second one is cognitive: studies on the analogical associations and on how these associations form patterns and schemes can contribute to the insight on how our brain works. The third one is didactical, since the learning of languages can greatly benefit from the knowledge on such patterns of association and their operation.

Some researchers that have studied this field are Adam Albright, Marco Battista, James Blevins, Hélène Giraudo, Fabio Montermini, Vito Pirrelli, Gregory Stump, who based themselves on the founding works of Peter Matthews, Martin Maiden, and Mark Aronoff, in the Word-and-Paradigm theoretical model.

The practical approach of these researches has the goal of analysing the paradigmatical structure of inflection, that is, to decompose the paradigm in

zones where the forms are realized on possibly distinct basic stems, and to examine the formal relations (on the phonological level) between these basic stems, studying the chains of predictability that permit us to handle both regular and irregular lexemes.

In chapter 1, I will outline some theoretical models and approaches, and their principal characteristics in relation to inflectional morphology.

In chapter 2, I will describe the works of some researchers that took on the task of analysing the paradigmatic distribution of inflection, mainly in Romance verbs.

In chapter 3, I will introduce a process of computation whose goal is an automated study of the relations between Italian verbal finite forms. To this sake it was necessary to develop a new program based on algorithm which can automatically find out the relations within the verbal paradigms. The program has been created by me, using the Java language. I defined a corpus of 136 verbs, comprising all schemes of conjugation, and analyzed and compared them.

Especially paragraph 3.10 shows an extract of the final result of the computation: a listing of formal relations, for each pair of paradigm cells, sorted by the number of lexemes that obeys them.

All the computed lexeme stems are shown as appendix in chapter 4.

Some terms recurring throughout the present work are defined as appendix on page 131.

The conclusions on page 133 close the present work.

1 Models and approaches

In this chapter I will describe several criteria of classifying morphological analysis.

Bybee (1988) describes three morphological models, Item and Process, Item and Arrangement, and Word and Paradigm, integrating Hockett's (1954) analysis of the first two models.

Stump (1991) classifies theories of inflection along two axes: *lexical* versus *inferential* theories, and *incremental* versus *realizational* theories.

Blevins (2006) distinguishes morphological models according to their morphotactical attitude: the two classes of models are *word-based* versus *root-based* or *morph-based* models. He then presents a classification of approaches according to the division between *abstractive* and *constructive* approaches.

1.1 ITEM AND PROCESS

The distinctive feature of Item and Process model is on the relation between underlying forms and surface forms. The model postulates a single underlying form, stored in the lexicon, for every lexeme and a series of realizational rules that transform its phonological and morphosyntactic features into a surface form.

	SEDERE ‘to sit’	
Person	Singular	Plural
1st	'sjɛdo	se'djamo
2nd	'sjɛdi	se'dete
3rd	'sjɛde	'sjɛdono

Table 1: Present indicative of SEDERE ‘to sit’.

Table 1 above shows the conjugation of Italian verb SEDERE, ‘to sit’, for the Present Indicative. It is immediately clear how the alternating forms of the stem are selected depending on the stress position.

According to the Item and Process model, the lexicon component of language contains one single form corresponding to the lexeme SEDERE. The form may be one of the surface forms or even a different form. A series of rules, found in another component of language, convert that underlying form to the needed surface form, modifying both its phonological and morphosyntactical features.

In the Item and Process model, the lexicon contains a relatively small inventory of forms, and if we conceive the lexicon component of language as a “storage space”, very little space is used.

However, the rules that produce the alternation shown above for the verb SEDERE only apply to a small set of Italian verbs, because the diphthongization of [e] → [jɛ] (and [o] → [wɔ]) under stress position is not productive anymore in Italian. On the contrary, most verbs underwent or are undergoing regularization, in a rate appreciable along the 20th century: [mo'vjamo] → [mwo'vjamo], [ko'tʃamo] → [kwo'tʃamo], or never had

alternation ['pɛrde] vs. *['pjɛrde]. The transformation rules, in the Item and Process model, "fail to distinguish between productive and non-productive alternations, between morphologically and phonetically conditioned alternations, and [… the model] allows underlying forms to differ too radically from the surface forms" (Bybee 1988, 120).

Since Item and Paradigm model derives all surface forms from a single underlying form, there is no way to account for relations between forms, or paradigmatic relations. For instance, relations between each surface form and the underlying form cannot explain the regularity of paradigmatic distribution of allomorphy (see § 2.1).

1.2 ITEM AND ARRANGEMENT

In the Item and Arrangement model, surface forms are built by morphs, in a certain succession (the arrangement). Morphs are surface forms of a more abstract object, the morpheme, which is stored in the lexicon. Alternating morphs, in particular, are stored along with the information on their distribution. Morphemes are the basic units of morphology, thus we can say that Item and Arrangement is a morpheme-based model. Every morpheme constituting a word has the same status: morphology does not distinguish between radical and inflectional elements. The correspondence between form (surface form) and function (morphosyntactic traits of which the morpheme is the exponent) is strictly one to one. The morphs are sequentially arranged according to the syntactic structure: there is no

separation between syntax and morphology; every part of the discourse is built with the same strategy.

The advantages of this model are, as outlined by Matthews (1972), its simplicity and homogeneity: only one basic unit is necessary, there is only one relation between a set of morphosyntactic features and an element of exponence, and only one structure serves to describe the grammar from word-form to sentence. Moreover, the Item and Arrangement model assumes that the hearer can isolate segments within the word, recognize morphs and associate them with the morpheme's syntactic properties.

Matthews (1972) convincingly addresses the above points with an extensive analysis (cf. 1.3 below).

For instance, he examines the case of the alternation between Latin imperfective and perfective forms of Latin verb RUMPO 'to break': *rump / rūp*. The alternation can be explained with the introduction of a zero morph as an allomorph of Perfective and [ru:p] as an allomorph of RUMP-: RUMP- + Perfective → [ru:p] + ø; but this violates the spirit of the model: "that each grammatical element should have a distinct and overt realisation". Another explanation of this alternation rests on the introduction of *morphophonemic operators* that are associated with the rules: RUMP- + Imperfective → *rup* + *m*<, RUMP + Perfective → *rup* + *L*, where "<" is the operator that puts the segment [m] before the previous segment [p], and "L" is the operator that lengthens the antecedent vocalic segment [u]. These operators are "quite unlike any genuine phonological element; they are

nothing more, in effect, than a notational device which will 'trigger off' the rules which we have stated. As such they are descriptively quite superfluous. There seems no point in saying that Perfective has an 'allomorph' L […], or that Imperfective has an 'allomorph' m< […], if the operations could be 'triggered off' directly by the morphological elements themselves" (Matthews 1972, 63).

1.3 WORD AND PARADIGM

While Item and Arrangement and Item and Process models reduce every construction, from the inflected word to the complete sentence, to arrangement of items or processed applied to them, Word and Paradigm approach focuses only on morphology, specifically inflectional morphology. As such, this approach deals mainly with morphologically complex (fusional) languages, be they complex in their nominal system, in their verbal system, or both.

In Word and Paradigm morphology, words are the basic units of syntactic structure: it can be classified as the typical word-based approach (cf. 1.6 below). In this approach, morphology and syntax are independent components of grammar, as outlined by Matthews (1972, 12):

> the task of generating the totality of sentences in the language could be divided into two major parts. On the one hand, the rules of syntax would specify a formula for each possible sentence (and consequently for each possible verb-form) […]: thus the sentence *puella moritur* might be represented, at this point, by some abstract formula of the

> kind 'Subject: the Nominative Singular of PUELLA; Verb; the 3rd Singular Present Indicative of MORIOR'. The rules of morphology, on the other hand, would then assign to each such formula its appropriate phonological shape: thus the shape *puella* would be assigned to the partial formula 'the Nominative Singular of PUELLA', and so on.

I would like to add that although morphology and syntax are independent components of grammar, their frontier is somewhat permeable. In fact, we can easily observe this permeability in diachronic change: while in Latin the Future Indicative is synthetic (*amabit*), in Proto-romance it is analytic (*amare habet*), that is, at some point the task of expressing the future tense was taken over by the syntactic component while the morphological component relinquished it. The inverse process happened in Romance languages, where the analytic forms collapsed into synthetic forms and re-entered the domain of inflectional morphology (*amerà, aimera, amará*)[1].

Another important aspect of the difference between sentence structure and word structure is in their relation with linearity, which is necessary in the former and both unnecessary and inconvenient for the latter: unnecessary because the ordering of elements within the word is fixed and never contrastive in character (Matthews 1972, 98) and inconvenient because the

[1] It is true that some phenomena, namely the "infixation" of clitic pronouns in future and conditional forms in Portuguese (e. g. *amaremos* + *o* → *amaloemos*), can be considered to be on the border line between morphology and syntax, but these belong to a class of phenomena concerning clitics ascribed to morphology (cf. Bonami and Boyé 2007).

ordering would be arbitrary in the typical case of cumulation (Matthews 1972, 72 foll.).

1.4 LEXICAL VERSUS INFERENTIAL THEORIES

Stump's analysis (2001) of theories of inflection divides these theories into *lexical* and *inferential*. His analysis is focused on morphosyntactic exponents, or inflectional markings. In lexical theories, the associations between morphosyntactic properties and aspect of a form's morphology – or exponents – are listed in the lexicon. In other words, affixes in the lexicon occur with a set of properties. In inferential theories, the same associations are identified by rules, which relate the inflected form with the root, and are selected by their morphosyntactic properties.

The combinations of lexical vs. inferential approach and incremental vs. realizational approach will be discussed, with examples, in the next paragraph.

1.5 INCREMENTAL VERSUS REALIZATIONAL THEORIES

Another division studied by Stump (2001), after the lexical versus inferential distinction, and independent from that, is into *incremental* and *realizational* theories. According to incremental theories, every exponent of morphosyntactic properties adds those properties to the word. While in realizational theories, it's the morphosyntactic properties of the word that select the exponents that realize these properties.

The combination of these classes, incremental and realizational, and the classes discussed above, lexical and inferential, yields four types of inflectional theories, exemplified by Stump in:

lexical-incremental: Lieber's model (Lieber 1992),

> In Lieber's theory, an affix's lexical entry is assumed to supply a subcategorization restriction limiting the kinds of contexts into which that affix might be inserted; for instance, the lexical entry of *-s* might be assumed to supply the restriction '[V_{stem} ____]' (= 'combines with a preceding verb stem'). As an affix joins with a stem, the morphosyntactic properties of the resulting whole are computed from those of its parts by a percolation mechanism; thus, *likes* acquires its syntactic category from its stem *like* and acquires the properties '3sg subject agreement', 'present tense', and 'indicative mood' from the suffix *-s*.

lexical-realizational: Distributed Morphology (Halle and Marantz 1993),

> Halle and Marantz assume that rules of syntax construct hierarchical combinations of abstract 'morphemes' (sets of morphosyntactic properties) into which concrete formatives are inserted from the lexicon; in order for a lexically listed formative X to be inserted into a morpheme Y, the set of morphosyntactic properties associated with X must be a subset of those constituting Y. On this view, the syntax is assumed to supply an abstract structure [V Y] (where Y comprises the properties '3sg subject agreement', 'present tense', and 'indicative mood'); *-s* is then insertable into Y because the morphosyntactic

> properties specified in its lexical entry aren't distinct from those constituting Y.

inferential-incremental: Articulated Morphology (Steele 1995),

> [According to Articulated Morphology], morphological rules effect changes in both the form and the content of the expressions to which they apply. For instance, *likes* arises by means of a rule applying to verb stems which are unspecified for subject agreement, tense, and mood; the application of this rule to a verb stem X results in (a) the addition of the suffix *-s* to X and (b) the addition of the morphosyntactic properties '3sg subject agreement', 'present tense', and 'indicative mood' to X's property set.

inferential-realizational: Word and Paradigm (Matthews 1972; Zwicky 1985; Anderson 1992),

> [In] Word-and-Paradigm theories of inflection […], an inflected word's association with a particular set of morphosyntactic properties licenses the application of rules determining the word's inflectional form; *likes*, for example, arises by means of a rule appending *-s* to any verb stem associated with the properties '3sg subject agreement', 'present tense', and 'indicative mood'.

1.6 WORD-BASED MODELS VERSUS ROOT-BASED OR MORPH-BASED MODELS

The morphotactical classification of morphological models deals with the status accorded to morphs versus full word forms. For word-based models,

the basic unit of a morphological system is the surface word form. Smaller parts that constitute a word (roots, stems, exponents) are the result of abstraction over sets of full forms. For root-based models, the basic units are morphotactical minimal forms, the morphs; the surface word forms are built from these basic units, which are associated with morphosyntactical properties, by force of rules of formation listed in the speaker's grammar.

1.7 ABSTRACTIVE VERSUS CONSTRUCTIVE PERSPECTIVE

Blevins (2006) defines two types of perspective: *abstractive* and *constructive*. Working in an abstractive perspective, a model treats surface word forms as the basic units; and smaller parts as roots, stems, and exponents are derived from the surface word forms in a process of abstraction. Under a constructive perspective, on the other hand, morphs are the basic units that build the surface forms.

Blevins' classification in *abstractive* versus *constructive* approaches is not fully independent from the previous one. In fact, he assigns the abstractive approach to word-based models, and the constructive approach to morph-based (or root-based or stem-based) models.

Item and Arrangement, Item and Process, and Word and Paradigm models can be interpreted constructively, in that the analysis determines basic units that are 'minimum meaningful elements' and describes the rules by which they are combined (Blevins 2006, 534). Even realizational models, like Aronoff's (1994), Anderson's (1992) and Stump's (2001) are constructive since the surface word forms are derived from minimal forms by means of

realizational rules. Blevins maintains that constructive approaches ultimately derive each surface form in isolation: once the derivation rules are found, the other members of the paradigm are ignored; and that this point of view is contradicted by psychological evidence.

According to the abstractive view, by contrast, the grammar *is* a set of relations among full surface forms and the stem is an abstraction emerging from these relations. Word and Paradigm models can have this perspective, namely those of Robins, Hockett and Matthews (Blevins 2006, 536): units like roots and exponents are always contingent on actual word forms. Also any segmentation into morphs, typical of post-Bloomfieldian linguistics, can be performed in an abstractive perspective, when those morphs are considered descriptive units rather than basic units stored in the lexicon.

An important aspect emerges from the relation, between basic units and surface forms, that is the focus of analysis in abstractive and constructive methods. Constructive methods focus on the relation of *derivation*, while abstractive methods focus on the relation of *predictability*.

1.8 A-MORPHOUS MORPHOLOGY

Anderson's work (1992) focuses on giving a model of morphology as a study on relations that occur between words, or surface inflected forms. His theory is not based on morphemes, hence the name of the theory.

Based on the assumption that a morph is a Saussurian sign, composing form (*signifiant*) and meaning (*signifié*), Anderson examines some evidence that

undermines the one-to-one correspondence between form and meaning: the case of infixes, where a breaking of a word into morphemes cannot explain the formation of Latin *rumpo*, from the root √*rup*, where the morph corresponding to the root would be interrupted by the morph corresponding to the nasal infix. The case of circumfixes: the expression of a single meaning with a discontinuous form, consisting in suffixes and prefixes that cannot be result of independent processing; as in the Indonesian *kebisaan* 'capability' from *bisa* 'to be able', where the two affixes *ke-* and *-an* do not occur in isolation. The case of empty morphs, where form is not associated with meaning, as for the Romance verbal thematic vowels, that do not mark any morphosyntactic category; or superfluous morphs whose meaning is irrelevant to the expression of morphosyntactic features required by the formation, as for *-th-* in the English verbs *lengthen* and *strengthen*. The case of cumulative morphs, that associate one form with many morphosyntatic features, as the Latin *-ō* in *ferō* 'I carry' is marking the subject as first person, in opposition to *fers* 'you carry', singular, in opposition to *ferimus* 'we carry', and present, in opposition to *feram*, 'I will carry'; or portmanteau morphs as in the French *du* 'of the' which combines *de* and *le* (all the examples above are from Anderson 1992).

Anderson suggests that a descriptive model should not reduce morphology to affixation, since non-affixational processes are part of human language: a morphological theory "should admit both affixational and non-affixional rules"(Anderson 1992, 68). Embracing Aronoff's view, word formation

rules relate one word to another, modifying the phonological form according to syntactic requirements. Also, focusing as stems as the basic units explains why diachronic semantic shift can occur on single words, not in morphemes, as is the case of *appreciable* vs. *appreciate*, where the meaning 'substantial' is not the result of the suffix *-able* being applied to the stem of *appreciate* 'to value'.

Anderson acknowledges *A-Morphous Morphology* to be an *Item and Process model* (according to Hockett's (1954) typology of morphological models), but supported by a *Word and Paradigm* view in its approach with inflection.

In A-Morphous Morphology the interface between morphology and syntax is represented by the inflectional properties of a word form as a whole, and does not derive from a morphological structure of the word.

1.9 PARADIGM FUNCTION MORPHOLOGY

Paradigm Function Morphology is a theory developed by Gregory T. Stump (1991) in a Word and Paradigm framework. Stump's starting point is the belief that an inflectional theory should base its distinctions on the empirical evidence. Moreover, that the morphological component of language is autonomous. Stump motivates his orientation towards an inferential-realizational theory with three points:

– the distinction of concatenative (affixational) and non-concatenative morphology: instead of explaining non-concatenative morphology as a

result of readjustment rules, needed by incremental theories, he assumes that "there is no theoretically significant difference between concatenative and non-concatenative inflection" (Stump 2001, 9). The rules that are associated with any set of morphological properties can thus be either concatenative or non-concatenative.

– Lexical theories associate an affix with properties that outline the affix *content*, and properties that establish (by restriction) the *context* where the affix should apply. As some models express that, content and context are the output and input of morphosyntactic rules. The problem, according to Stump, is that lexical theories need to choose when a property belongs to the rule's content or context, as is exemplified by the Bulgarian affix *-m* (Stump 2001, 11):

> In Bulgarian, there is a class of verbs […], which exhibit a special suffix *-m* in the first person singular of the present tense (e.g. *davam* 'I give'). The question is: is *-m* a 1sg suffix which subcategorizes for a present-tense stem, or is it simply a 1sg present-tense suffix? That is, is it an exponent of present tense, or is it simply restricted to the context of forms that are present in tense? Lexical theories demand that a choice be made, but the choice is inevitably an arbitrary one. In lexical theories, the need to choose in such cases is an artifact of the assumption that affixes are inserted from the lexicon and may therefore subcategorize for particular types of contexts: in Articulated Morphology, the need to choose is an artifact of the assumption that inflectional morphology is information-increasing.

– The representation of inflectional morphology in lexical theories is the same as the representation of the phrase structure: a "branching structure of hierarchically organized constituents". Stump maintains that this structure is actually needed only for compound words, and adopts the assumption held by inferential theories that

> An uncompounded word's morphological form is not distinct from its phonological form.

Stump (2001, 43) defines PARADIGM of a lexeme L as a set of CELLS; a CELL <Y,σ> is the association of an inflected form Y (a phonological string) of L with a set of morphosyntactic properties σ. A set of morphosyntactic properties identifies a cell; an inflected form occupies a cell.

A PARADIGM FUNCTION transforms a paring of a root X and a morphosyntactic property set σ into a cell of the paradigm:

$$PF(X,\sigma) \rightarrow \langle Y,\sigma \rangle$$

It is worth noting that the basic unit that is input to the paradigm function is the root rather than the lexeme: this encodes formally the possibility of lexemes to have allomorphy as multiple entries with multiple roots in the lexicon.

Paradigm functions are defined as sequences of REALIZATION RULES which, to summarize, are selected using Pāṇini's principle –that a rule that applies to a more specific set of parameters overrides a rule that applies on a

more general level, cf. (Stump 2001, 23) and (Anderson 1992, 132)– by morphosyntactic features and lexeme properties.

Some realization rules can be rules of referral, when they assume as their own output the output from another realization rule: this key concept handles cases of syncretism, where different cells of a paradigm are systematically occupied by the same phonological form.

Paradigm Function Morphology has been extended by Bonami and Stump (to appear) to include a stem function that provides phonological matter to the paradigm function:

$$PF(<L,\sigma>) = PF(Stem((<L,\sigma>)).$$

If we assume that the stem function provides not necessarily one, but possibly a set of several inputs to the paradigm function, the model can handle the case of overabundance, that is when a paradigm cell for a lexeme (or for the entire language) can be occupied by more than one form, as the Italian first person singular present indicative of SEDERE 'to sit' can be either *siedo* or *seggo*.

The set of all paradigm function forms the language's inflectional morphology.

1.10 NETWORK MORPHOLOGY

Network Morphology is a lexeme-based theoretical framework, developed by Corbett and Fraser (1993), classifiable as a Word and Paradigm approach. Its generalizations about the morphological structure are obtained studying

the paradigmatic organization, that is, following Stump's (2001) classification, it falls under the inferential-realizational models.

Network Morphology recognizes the morphological component of language as a separate grammatical component (Brown and Hippisley 2012).

The key point of Network Morphology is that every item of analysis (a node) is connected to other items through a network of relations of inheritance. Every node inherits features or provides new features. This is reflected by the formalization, that is expressed by means of the DATR language (Evans and Gazdar 1996). In fact, Network Morphology being a strictly formal framework, the analysis is fully computable. A node contains data in the form of key and value pairs. The obligatory entry in the node's data is the link to the node's parent, from which the node inherits defaults: that is, properties not expressed in the node itself. Other data can be on the syntactic level, as the syntactic category of a node, on the semantic level, as the gloss, on phonological level, as a string of phonological segments forming a stem, or on morphological level, as the conjugation class information.

This is the example for the entry of the Latin verbs HORTOR and UTOR, from Brown and Hippisley (2012, 235):

Hortor:		Ūtor:	
<>	== DEPONENT	<>	== DEPONENT
<gloss>	== encourage	<gloss>	== use
<root>	== hort	<root>	== ūt
<stem>	== CONJ_1.	<stem 3>	== ūs
		<stem>	== CONJ_3.

These two nodes inherit every property from the DEPONENT node, except of course the properties here listed. “Hortor” will inherit is morphology from the more general node CONJ_1, which will supposedly contain the relation between the inflected forms and the stems, and default relations between the stems and the root. “Ūtor” derives its morphology from the node CONJ_3, which will have information analogous to the CONJ_1 node; but the form of the <stem 3> property is listed here, rather than be derived with the default rules contained in the CONJ_3 node.

2 Practical approaches

In this section I will introduce some recent works on the problem of finding the paradigmatic structure of inflection.

From a theoretical point of view, the trend common to these researches in inflectional morphology is to adopt an inferential-realizational approach, producing word-based models in an abstractive perspective, summarised in Table 2:

inferential approach (see §1.4)	Associations between morphosyntactic properties and aspects of a form's morphology are identified by rules associated with those properties.
realizational approach (see §1.5)	The morphosyntactic properties of a word identify the exponents that realize these properties.
word-based model (see §1.6)	Smaller parts that constitute a word (roots, stems, exponents) are the result of an operation of abstraction over sets of full forms.
abstractive perspective (see §1.7)	The grammar is a set of relations among full surface forms; and units like stems and exponents are contingent abstractions emerging from these relations.

Table 2: trends of research in inflectional morphology

Pirrelli and Battista (2000) carry out an analysis of alternations in Italian verbs and examine in detail the phonological processes involved and the

distribution of these alternations (see § 2.1). Their work focuses on a detailed analysis of stem allomorphy, from a purely morphological perspective, stemming from the tradition of Matthews (1974), Stump (1991), Aronoff (1994).

Albright (2002) has developed computational processes for the identification of basic stems in inflectional systems (see § 2.2).

Bonami and Boyé (2001; 2003; 2014) perform an analysis of alternations in French verbs also with a paradigmatic approach, and in the latter work (2014) with quantitative methods based on the concept of interpredictability of surface forms (see § 2.3).

2.1 PIRRELLI AND BATTISTA'S WORK

Pirrelli and Battista's work (2000) shows how the distribution of allomorphy in Italian conjugation can be described on a level of abstraction that is more coherent and satisfactory than traditional syntagmatic accounts, where explication of surface forms is based on local phonological transformations.

Pirrelli and Battista proceed analysing the alternations found in Italian irregular verbs, and enumerating all the phonological processes encountered, e.g. velar palatalization, [g]-insertion, [r]-drop, palatal lengthening, ablauting, etc.

Pirrelli and Battista highlight some important issues that are hardly explained with phonological constraints only (Pirrelli and Battista 2000, 351):

- Rule ordering inconsistencies: it is not possible to find a coherent ordering of phonological rules to account for thematic vowel elision and palatalization, or for diphthongization and consonant gemination.
- Absence of overt phonological motivation: it is the case of augment insertion (or deletion), or consonant gemination.
- Number of exceptions: once a rule for phonological processes, as palatalization or diphthongization is found for a small sample of lexemes, many more are found which do not undergo the same processes under the same context.
- Distributional correlation between phonological and morphomic alternations: suppletive alternation shows the same distribution as alternations phonologically motivated.

The paradigm partition classes are summarised in Table 3 below, where each paradigm cell is associated with an indexed basic stem S to S_8:

	singular			*plural*		
	1	*2*	*3*	*1*	*2*	*3*
Present subjunctive	S_2			S_4		S_2
Present indicative	S_2	S_3		S_4	S	S_2
Imperfect indicative	S					
Imperfect subjunctive						
Past simple	S_5	S	S_5	S		S_5
Present imperative	–	S_3	S_2	S_4	S	S_2
Present conditional	S_6					
Future indicative						
Present gerund	S					
Present participle						
Past participle	S_7					
Present infinitive	S_8					

Table 3: distribution of basic stems in Italian (Pirrelli and Battista 2000, 359)

Following Pirrelli and Battista's analysis, partition classes have a systematic distribution across lexemes; they do not correlate with morphosyntactic features, meeting Aronoff's definition of morphomic relations; they explain underapplication of phonological rules (the solecism [v'ɛngino] (subjunctive present 2pl. of VENIRE 'to come') *versus* *[v'ɛndʒino]), overapplication of phonological rules ([kreʃʃ'uto] (past participle of CRESCERE 'to grow') *versus* *[kresk'uto]), the distribution of suppletive stems and the distribution of doublets (Pirrelli and Battista 2000, 367).

It is worth noting here, that the regular and coherent schema proposed by Pirrelli and Battista precludes eight highly irregular verbs: ANDARE, AVERE, DARE, DIRE, ESSERE, FARE, SAPERE, STARE.

In Pirrelli and Battista's approach, the basic stems are all different. That is equivalent to saying that not all verbs use all the basic stems S to S_8, but only a subset of them is attested for any Italian verb (the verb that uses most of them is DOLERE with 6 distinct basic stems); the choice of a basic stem for a given paradigm cell is subsequent to an operation of re-indexing. For example, the verb CONOSCERE has a re-indexing of $S_4 \rightarrow S$, $S_2 \rightarrow S$; PIACERE has a re-indexing of $S_4 \rightarrow S_2$, $S_3 \rightarrow S$. I prefer to consider the basic stems distinct even when they are formally identical and to introduce an operation of identity in the number of operations that relate a basic stem to another, like velar insertion, gemination, diphthongization, etc.

Subsequent articles (Giraudo, Montermini, and Pirrelli 2008; Montermini and Boyé 2012; Montermini and Bonami 2013) illustrate the relations by default of Italian basic stems and give a formal definition of lexeme regularity (Montermini and Bonami 2013, 13): "A regular lexeme is one for which all stems are linked by predictable relations, an irregular one is one for which at least one stem is not linked to the others by such a relation".

Figure 1 below shows the paradigm structure of Italian verbs. The lines connecting the basic stems point out the relations of maximum interpredictability. The stems are T1 (of e.g. Imperfect), T2 (Present

Subjunctive 1s), T3 (Indicative Present 2s), T4 (Indicative Present 1p), T5 (Indicative Past Simple 1s), T6 (Future), T7 (Past Participle), T8 (Infinitive).

Figure 2 shows the paradigm of a "regular" verb of group "*-ere*, proparoxytone" with the stem relations shared by all regular verbs of the same group.

Figure 3 and Figure 4 show the structure of two "irregular" verbs of the same conjugation group, where some basic stems are not linked by predictable relations (*-v-* ↔ *-ss-*, *-f-* ↔ *-bb-*).

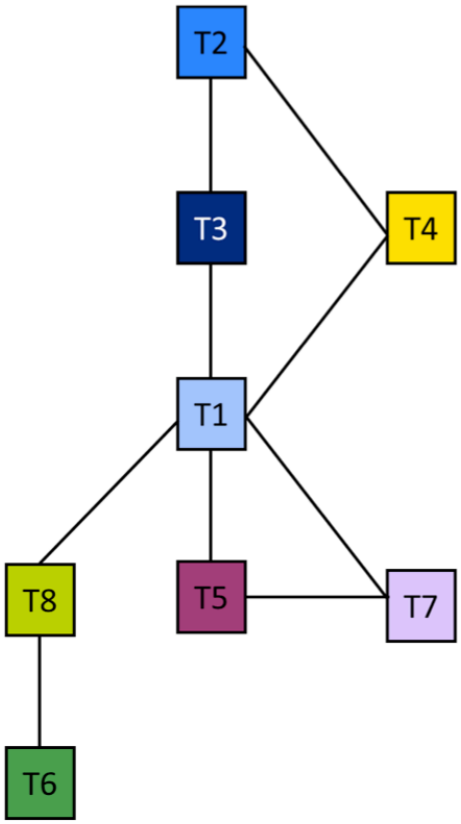


Figure 1: the paradigm structure of Italian verbs (Giraudo, Montermini, and Pirrelli 2008, 7)

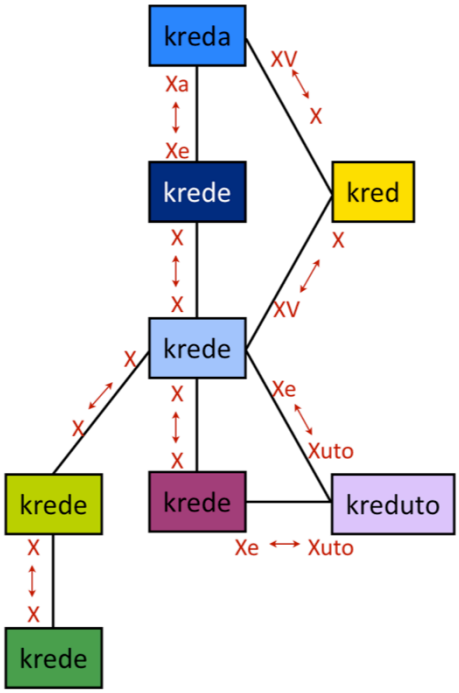


Figure 2: the paradigm structure of a "regular" verb, CREDERE 'to believe' (Giraudo, Montermini, and Pirrelli 2008)

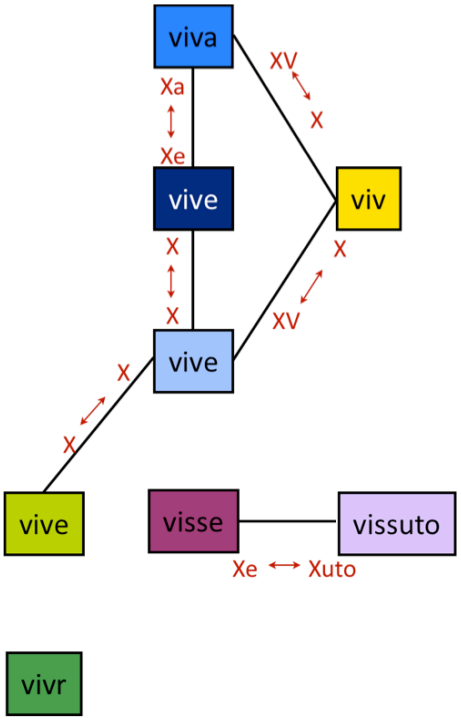


Figure 3: the paradigm structure of an "irregular" verb, VIVERE 'to live' (Giraudo, Montermini, and Pirrelli 2008)

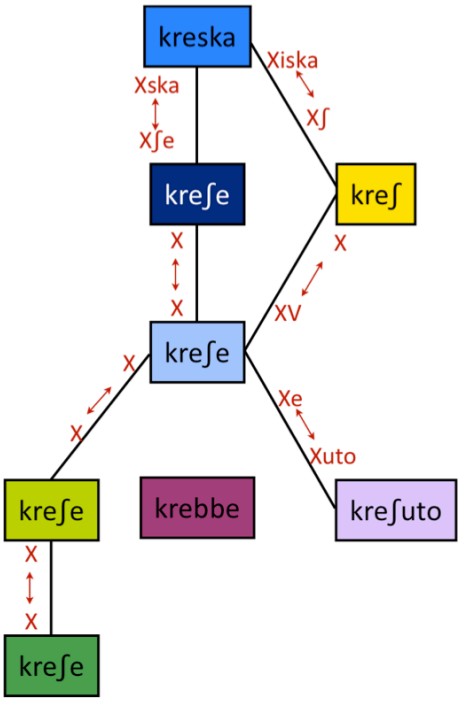


Figure 4: the paradigm structure of an "irregular" verb, CRESCERE 'to grow' (Giraudo, Montermini, and Pirrelli 2008)

Table 4 below lists the eight more frequent patterns of alternation in Italian conjugation:

S_1 [2]	S_2	S_3	S_7	n. lexemes[3]	example
Xa	Xi	Xa	Xat	1504	LAVARE
Xi	Xiʃʃe	Xiska	Xit	135	FINIRE
Xi	Xe	Xa	Xit	32	SENTIRE
Xnde	Xnde	Xnda	Xzo	32	PRENDERE
Xdʒe	Xdʒe	Xga	Xt	30	SPINGERE
Xde	Xde	Xda	Xz	29	CHIUDERE
Xne	Xne	Xnga	Xst	18	PORRE
Xette	Xette	Xetta	Xess	17	METTERE

Table 4: most frequent patterns of alternation in Italian (Montermini and Bonami 2013, 13)

[2] S_1 here corresponds to S in Pirrelli.

[3] From LIP (*Lessico di frequenza dell'Italiano parlato*, De Mauro et al. 1993) corpus.

[3] From LIP (*Lessico di frequenza dell'Italiano parlato*, De Mauro et al.

The ranking of formal relations between basic stems, based on the frequency of the lexemes involved in each, is an empirical solution; and to know precisely what bearing it has in the speaker's competence would require, among others, psycholinguistic studies, as the authors acknowledge.

2.2 ALBRIGHT'S WORK

Albright (2002) developed a computational algorithm to extract bases by comparing surface forms, called the Minimal Generalization Learner.

Even if we accept that an individual memorizes every inflected form ever heard (or read), there are inflectional system so complex that it is not possible to memorize (or build beforehand during a learning stage) every variation of a single lexeme: for example, in the language Shona, a Bantu language, the verb has so many variants for mood, tense, aspect, implication, and for concordance of subject, object, etc., that the inventory of forms for a lexeme can reach the number of sixteen thousand billions (Albright 2002, 1; Fortune 1955, 238 foll.). Yet a speaker must be able to produce every form needed and comprehend every form encountered; this means that he must use a strategy of rules (be these encoded in a grammar or emerging from analogy) to derive a form from another form.

Continuing the reasoning above, and since the pairwise relation between couples of forms could also be too many to learn, there must be a path of

relations connecting a preferred base form to other forms. In Albright's work, these hypotheses are proposed:

- There is a single base for every lexeme;
- The base is fixed: that is, it's the realisation of a fixed paradigm slot, the same for all lexemes;
- The base is the form that preserves the most contrast: that is, whose number of segments hidden by neutralisation is minimal;
- The learners impose a structure on paradigms, learning a path of relations from the base to any given form.

With these assumptions, the Minimal Generalization Learner algorithm is presented. With two forms as input, a result is provided as output with the rule X→Y/A_B. When another pair of forms is provided as input, the algorithm matches the variables (X and Y) and searches for shared features in the context (A and B in the formula above) of the previous rule found and the new one. If shared features are found, the original rule is generalized so that the context is restricted to them. If shared features are not found, or the variables do not match, another rule is spawned.

For instance, we have the following excerpt from a hypothetical paradigm:

absolutive	*ergative*
lag	lagi
basag	basagi
rag	ragu
tip	tipu

Table 5: two cases of an hypothetical declension (Albright 2002, 38)

The pair <lag, lagi> yields the rule

1) Ø→i/lag_#.

The pair <basag, basagi> yields

2) Ø→i/basag_#.

The comparison of these rules gives the generalised rule

3) Ø→i/[+cons, +cont, +cor, etc.]ag_#,

and the rules 1) and 2) are removed.

The third pair <rag, ragu> gives the output

4) Ø→u/rag_#.

The variables of this rule do not match with 3), so the new rule is not merged. Finally, the fourth pair <tip, tipu> gives:

5) Ø→u/tip_#,

whose variables match 4), so 4) and 5) are removed and generalised to:

6) Ø→u/[-continuant,-coronal,-lateral, etc.]_#.

Finally, for each rule (3 and 6 in the small example above) a reliability score is computed, equal to the number of forms that the rule correctly derives divided by the number of forms that the rule can apply to. The best candidates as bases are those forms, which the most reliable rules transform into other forms of the paradigm.

To summarize, Albright's computational model's characteristics are:

- The method gives quantitative results as to how much of the inflectional data can be reproduced using the derived rules and how much cannot, and so must be considered irregular.
- The model can operate incrementally: more data can be provided as input to readjust the rules and their reliability score.

The downside, in my opinion, is that Albright's observations assume the single base hypothesis, that the base is single for every paradigm: an assumption that reduces computational complexity but is perhaps too restrictive, as Albright himself acknowledges (2002, 129): "When inputs are restricted to a single surface form, […] it is often impossible to find a single form that unambiguously reveals the surface properties of all of the members of the paradigm."

2.3 BONAMI AND BOYÉ'S WORK

Bonami and Boyé wrote a series of articles on paradigmatic distribution in French verbs and adjectives, based on the theoretical tradition of Matthews

(1972), Maiden (1992), and Aronoff (1994), and alongside the analyses of Brown (1998), Pirrelli and Battista (2000), Stump (2001), and Blevins (2003). Their earlier articles (Bonami and Boyé 2001; Bonami and Boyé 2003) carried out a qualitative analysis of paradigms and dependency relations between base stems in a hierarchical scheme. In later works (Bonami and Boyé 2014) they focus on the quantitative aspect of the analysis.

The work of Bonami and Boyé rests on the following hypotheses (Bonami and Boyé 2014, 2–3)[4]:

1. lexemes are not (necessarily) associated with a single stem, but to an indexed set of stems, called *stem space*,
2. the stem space of a lexeme induces a partition of the paradigm whose zones are not morphosyntactically coherent,
3. the same partition accounts for regular alternances of open lexeme classes as well as suppletion of isolate lexemes,
4. the rules accounting for the relations between cells of the stem space are independent from the rules accounting for the choice of a cell to find a required form of the lexeme,
5. inflection proper is invariable: out of rare instances of suppletion of a whole inflected form, all lexemes of a given category carry the same inflectional markings. In consequence, every apparent variation in inflectional markings is analysed as stem allomorphy,

[4] Translation mine.

6. the extent of stem spaces is fixed: every lexeme belonging to the same syntactic category have a stem space with the same structure,
7. the only relevant rules of stem formation are relations entailing by default. These relations connect two cells *i* and *j* of the stem space and state, on the basis of the stem's phonological properties, which morphological function connects the content of cell *i* to the content of the cell *j* for the lexemes of the same inflectional class.

The approach of (Bonami and Boyé 2014) introduces two strategies: first, the application of measures for inter-predictability relations between paradigm cells, following the method outlined in (Ackerman, Blevins, and Malouf 2009); second, adopting the abstractive perspective of *deducing* a form from the other forms of the paradigm, rather than *building* a form from an abstract representation (see Blevins' (2006) classification of abstractive and constructive approaches in §1.7, page 16).

Ackerman et al. describe a proceeding already introduced in (Moscoso del Prado Martín 2003; P. Milin et al. 2009) where the predictability of the relation between two form is associated with the measure of *entropy*. Entropy is a concept borrowed from physics – and the explanation of its meaning in physics lies outside the scope of the present work[5] – but a key

[5] I will try to illustrate the concept of entropy in thermodinamics with an example. If we have a crystal of diamond at a very low temperature –near 0°K– its atoms vibrate but in a confined space and with low energy: a prediction about their properties, such as position in space, would approach the reality. If instead we have a very hot gas, its atoms would vibrate, move, spin around and collide with very high energy, and any prediction about

point is that Shannon (1948) transposed that concept to the theory of information.

Given a variable X, and $x_1 \ldots x_n$ the possible values that the variable X can assume, and $p(x_i)$ the probability that X could assume the value x_i, the entropy H(X) is the value:

$$H(X) = -\sum_i p(x_i) \log_2 p(x_i)$$

(the minus balances the fact that a logarithm of a probability is always negative). The entropy of the variable X is zero when only an outcome is possible with probability equal to 1 (or 100%), and approaches infinity when the number of outcomes approaches infinity while their probabilities approach zero. When there are two possible equiprobable outcomes for X (e.g. in a coin flip), the entropy of X equals 1, or, in other words, *the information carried by X is 1 bit*.

The *joint entropy* H(X,Y), of two variables X and Y, is linked to the probability $p(x_i,y_j)$ that X assume the value x_i and Y assume the value y_j:

$$H(X,Y) = -\sum_{i,j} p(x_i, y_j) \log_2 p(x_i, y_j);$$

and, finally, the *conditional entropy* H(Y|X) is the entropy of Y given the value of X, based on the probability $p(y_j \mid x_i)$ that Y assume the value y_j when X assumes the value x_i:

---

their properties would be extremely difficult. In the first case, a situation of relative *order*, the value of entropy is low; in the second case, a situation of relative *disorder*, the value of entropy is high.

$$H(Y|X) = -\sum_i p(x_i) \sum_j p(y_j|x_i) \log_2 p(y_j|x_i),$$

or, substituting for the equations above,

$$H(Y|X) = H(X,Y) - H(X).$$

The conditional entropy H(Y|X) can be interpreted as the *predictability* of Y on the basis of X, which brings us to its application in morphology. As Ackerman et al. state, given a paradigm "we can calculate the conditional entropy of any one cell given any other cell".

Ackerman et al. illustrate this with a sample from Saami nominal declension, namely the relation between nominative singular and locative plural forms. The suffixes for these forms are, respectively, Ø and -in. But the cells alternate between a strong form and weak form, e.g. *bihttá* 'weakening' → loc. pl. *bihtáin*, *baste* 'strengthening' → loc. pl. *basttiin*. The combinations of such alternations with their joint probabilities are:

Nom. Sg.	*Loc. Pl.*	*P*
strong	strong	0.0
strong	weak	0.5
weak	strong	0.5
weak	weak	0.0

Table 6: joint probabilities of alternations in two cases of Saami (Ackerman, Blevins, and Malouf 2009, 68)

From these data the conditional entropy is calculated as H(LOC.PL|NOM.SG) = 0, which means that given the nominative singular form, the locative plural form is always known with no uncertainty.

As far as French is concerned, Bonami and Boyé (2014) proceed by first identifying the inflectional markings of French verbal system according to the principle 5 (page 37). These inflectional markings are shown in Table 7 below:

	1s	2s	3s	1p	2p	3p
PRESENT	id.	id.	id.	-ɔ̃	-e	id.
IMPERATIVE	–	id.	–	-ɔ̃	-e	–
IMPERFECT	-ɛ	-ɛ	-ɛ	-jɔ̃	-je	-ɛ
SUBJUNCTIVE	id.	id.	id.	-jɔ̃	-je	id.
FUTURE	-ʁɛ	-ʁa	-ʁa	-ʁɔ̃	-ʁe	-ʁɔ̃
CONDITIONAL	-ʁɛ	-ʁɛ	-ʁɛ	-ʁjɔ̃	-ʁje	-ʁɛ
SIMPLE PAST	H	id.	id.	-m	-t	H-ʁ
PAST SUBJ.	-s	-s	id.	-sjɔ̃	-sje	-s

INFINITE	-ʁ	PAST. PART. M. S.	tronc_C
PRESENT PART.	-ɑ̃	PAST. PART. M. P.	tronc_C
		PAST. PART. F. S.	id.
		PAST. PART. F. P.	id.

Table 7: inflectional markings of French verb (Bonami and Boyé 2014, 6)

where "H" and "tronc_C" are morphonological functions that, respectively, change the vowel in verbs of the first conjugation, and truncate the last consonant. Also, it must be noted that the marking -ʁ of infinite is subject to the rule ʁ→Ø/e_# out of liaison environment.

Bonami and Boyé consider nine forms suppletive to limit the number of stem spaces (12 instead of 16) needed: *je suis* (vs. expected[6] *j'es*), *nous sommes* (vs. *nous étons*), etc. The purpose of listing of forms as suppletive can be seen as roughly corresponding to the exclusion of Italian lexemes ANDARE, AVERE, DARE, DIRE, ESSERE, FARE, SAPERE, STARE in Pirrelli et al. (cf. § 2.1).

The second step is the identification of transformation rules, for every alternating pair <φ,ψ> of forms minus the inflectional markings. The rules computed are of the form

$$\alpha \rightarrow \beta / \#X\_Y\#$$

where φ = XαY, ψ = XβY, X is maximal, and α is minimal; then the rules are submitted to Albright's Minimal Generalisation Learner method (cf. § 2.2).

Next, Bonami and Boyé group the forms in classes according the principle that φ and ψ are in the same class if and only if the set of rules that can be applied (according to the context restrictions identified by the Minimal Generalisation Learner) to φ is the same set as the rules that can be applied to ψ.

Then, the conditional probability of rules, pertaining to a given class of forms, is given by the number of forms instancing each rule (weighed

[6] Hypothetic forms that would be coherent with the relations found in all other lexemes.

against a French corpus, BDLEX), divided by the number of forms contained in that class (Bonami and Boyé 2014, 10).

Finally, the conditional entropy between pairs of paradigm cells is computed. Now, if we consider a partitioning of paradigm cells in zones where conditional entropy is null, we get a stem space of 15 zones, as in the following table (Bonami and Boyé 2014, 15):

	1s	*2s*	*3s*	*1p*	*2p*	*3p*
PRESENT	3a	3b	3b	1a	1b	2
IMPERFECT	1a	1a	1a	1c	1c	1a
IMPERATIVE	–	5	–	6	6	–
SUBJUNCTIVE	7	7	7	8	8	7
FUTURE	10	10	10	10	10	10
CONDITIONAL	10	10	10	10	10	10
SIMPLE PAST	11	11	11	11	11	11
PAST SUBJ.	11	11	11	11	11	11
INFINITE	9	PAST PART. M. S.				12
PRESENT PART.	4	PAST PART. M. P.				12
		PAST PART. F. S.				12
		PAST PART. F. P.				12

Table 8: French verbal paradigm partitioned in fifteen zones (Bonami and Boyé 2014, 16)

However, considering that very low entropy values (< 0.12) mean that only a very small number of verbs (1 to 5, as order of magnitude) require such a

fine-grained distinction, Bonami and Boyé apply a threshold of 0.01 of the conditional entropy value for considering a pair of cells as pertaining to the same stem space, reducing the stem space of French verbs to 7 zones:

	1s	*2s*	*3s*	*1p*	*2p*	*3p*
PRESENT	C	C	C	A	A	B
IMPERATIVE	–	C	–	A	A	–
IMPERFECT	A	A	A	A	A	A
SUBJUNCTIVE	B	B	B	A	A	B
FUTURE	D	D	D	D	D	D
CONDITIONAL	D	D	D	D	D	D
SIMPLE PAST	E	E	E	E	E	E
PAST SUBJ.	E	E	E	E	E	E

INFINITE	F	PAST PART. M. S.	G
PRESENT PART.	A	PAST PART. M. P.	G
		PAST PART. F. S.	G
		PAST PART. F. P.	G

Table 9: French verbal paradigm partitioned in seven zones (Bonami and Boyé 2014, 17)

3 Finding basic stems

In this chapter, I will describe the steps I performed in this morphological analysis of Italian verbs.

The first step consists in shearing off the phonological segments that are invariant across all the lexemes; these segments, very broadly, correspond in part to the traditional desinences, but no grouping in classes has been performed. Thence, the thematic vowel is left on the variant part. To accomplish this step, I propose here a computational method based on Levenshtein's algorithm, which in turn is an algorithm used to compute the editing distance (see §3.1) between strings of symbols. The implementation I programmed does not assume the direction of inflection or, in other words, is independent of the inflection being suffixal, prefixal or discontinuous.

The next step is calculating the exponent invariance, using the algorithm presented above; the results of this operation are summarised in Table 11 on page 61.

The next process consists in grouping the paradigm cells in sets such that every lexeme has the same stem in all cells of each set (or "metacell"). This partitions the paradigm in zones of maximum interpredictability since there is not internal variation of stems.

Finally, the relations between the stems corrisponding to pairs of metacells, are computed and listed. Since the complete listing would take several thousand lines of text, only an extract is presented below.

Working on these data I have encountered three main issues that required me to take decisions: defectiveness, overabundance, and the representation of stress.

Defective verbs are verbs for which some cells of the paradigm are empty: no form is attested for the particular morphosyntactic properties associated with that cell. For these cases, an "always-match" strategy has been chosen: an empty cell can be grouped with any other cell, any formal relation can link an empty cell to a non-empty cell.

Overabundant verbs are verbs for which a cell, or more cells, can contain multiple forms rather than a single form: this is the case, for example, of Indicative Present 1s of verb DOVERE (*devo, debbo*) or Subjunctive Present 1s (*deva, debba*). Both forms have been inserted in the computation, with the result of obtaining some relations that are not at all expected (*devo* → *debba*, *debbo* → *deva*). This issue deserves further reasoning.

Since stress is mobile in Italian, that is its position cannot be determined by the phonological context, it has to have a role in the computation. Yet it is not a proper phonological segment; stress cannot be added or removed, but only moved, and only one stress can be at any time in a word form. This seems a simple constraint for us, but for a computational algorithm it's not. I decided to break every form in two, the protonic and postonic parts, compute them separately, and then reunite them. This guarantees that the stress is ever present and unique.

3.1 MEASURES OF DISTANCE

The distance d between two elements x and y, members of a set E is the result (a real number) of a function D, such that:

$d = D(x,y);\ x,y \in E;\ d \in R.$

This is a completely general description giving way to a multitude of implementations, specifically implementations of the function D.

For example, we can measure the distance between two points on the earth's surface as Euclidean distance, that is measuring the segment of a straight line passing inside the earth's surface; or as the crow flies, along a curve parallel to the surface; or as a distance that has to be covered travelling along a certain road.

With analogous considerations, it's possible to devise several functions to compute the distance between two strings of symbols, or, inversely, their similarity degree.

One of these function is the Hamming distance (Hamming 1950), which is given by the number of symbols that differ, in the same position, between the two strings. In the example,

H("abcde","abdef") = 3,

only the symbols "a" and "b" are in the same position in both strings of five symbols, so the Hamming distance, computed for these two strings, is 3.

3.2 LEVENSHTEIN'S DISTANCE

Vladimir Levenshtein found the algorithm, known by his name (Levenshtein 1966), that measures the distance between two symbol strings, *x* and *y*, summing up the cost of basic operations needed to transform *x* into *y*. The basic operations are: insertion (of a symbol occurring in y but not in x), deletion (of a symbol occurring in x but not in y), substitution of a symbol; and finally the no-modification operation, in case the same symbol occurs in both strings in the expected position.

The cost of each of these operations, in the original implementation, is 0 for no-operation and 1 otherwise, as shown in Table 10 below:

Example	Distance	Transformation
"abc" → "abc"	0	No transformation necessary
"abc" → "axbc"	1	Insertion of "x"
"abc" → "ac"	1	Deletion of "b"
"abc" → "axc"	1	Substitution of "b" with "x"

Table 10: basic operations of Levenshtein's algorithm

Levenshtein's algorithm builds a matrix, with as many columns as the number of symbols in the starting string plus one, and as many rows as the number of symbols in the destination string plus one. The following table shows the matrix needed for computing the distance between the two strings "cane" and "care", initially empty:

•	•	c	a	n	e
•					
c					
a					
r					
e					

a zero is inserted into the top-left cell of the matrix. Each other cell in the first row receive a value which is the sum of the value of the cell at the left of it and the cost of a single deletion operation. Each other cell in the first column receive a value that is the sum of the value of the cell at the top of it and the cost of a single insertion operation.

•	•	c	a	n	e
•	0	1	2	3	4
c	1				
a	2				
r	3				
e	4				

Then, the rest of the matrix is filled inserting a value in each cell, proceeding from left to right and from top to bottom. If the corresponding symbols in the two strings (shown here at the top of the cell's column and at the left of the cell's row) are the same, the value from the cell at the top left corner of the new cell is simply copied. Otherwise, the value of the cell

above is summed with the cost of an insertion operation, the value of the cell at the left is summed with the cost of a deletion operation, and the value of the cell above at the left is summed with the cost of a substitution operation, and the minimum of these three values is copied into the new cell.

After filling all the cells, Levenshtein's algorithm has calculated the matrix:

•	•	c	a	n	e
•	0	1	2	3	4
c	1	0	1	2	3
a	2	1	0	1	2
r	3	2	1	1	2
e	4	3	2	2	1

where the value of the bottom right cell is the final result, the measure of Levenshtein's distance between the two given strings.

3.3 APPLICATIONS FOR LEVENSHTEIN'S DISTANCE

Levenshtein's distance algorithm is used in genetics, to compare sequences of nucleobases in DNA from individuals of the same species or even different species.

It is used in word processing programs to suggest corrections during spell checking, where the program presents the user the most likely few words to replace a wrongly-spelled word. It is also useful in optical character recognition, when a probable word (typically the result from a neural network analyzing the image) is tested against a lexicon.

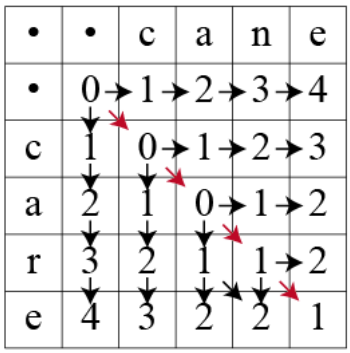
This distance algorithm is also used in forms, especially on the world wide web, when the user types a string not corresponding to a possible choice, for instance, if a user types “thrusday” for a weekday, the program on the website can compare the entry with a list of candidates, and automatically substitute the best choice when the distance measure is below a given threshold, in this case the correct “thursday”.

3.4 IMPLEMENTATION OF LEVENSHTEIN’S ALGORITHM

Studying Levenshtein’s distance, I observed that it could be desirable to take advantage of Levenshtein’s algorithm not only to find a single numeric value, but also to obtain a list of the effective basic operations used to transform a string into another, with the goal of classifying the morpho-phonological transformations occurring between any two verbal forms.

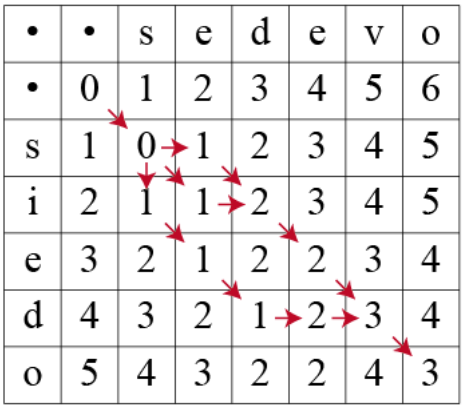
So I programmed a version of the algorithm, using Java language, that returns such list, or, precisely, all sequences of operations that bring to the same value, the distance measure. Indeed, if the distance measure is a unique and minimal value, just like the mathematical section of Levenshtein’s article shows, the sequence of basic operations that results in said value is not unique, as I will show here.

Now we go back to the example where I showed how to compute the distance between “cane” and “care”. In the following image I highlighted with arrows the origin of cell values, calculated in the way described above (§ 3.2, page 48).



Only one path brings to the final cell, and it's highlighted in red. For the case of these two strings, my program returns the value 1 and also the sequence: {N(c);N(a);S(n,r);N(e)}, meaning: keep "c", keep "a", substitute "n" with "r", keep "e".

But there are cases where more paths bring to the same value, as shown by the following matrix, where the algorithms compute the distance between "*sedevo*" and "*siedo*".



In this case, we get three paths which bring to the final result, 3, and they are:

N(s);D(e);S(d,i);N(e);S(v,d);N(o),

N(s);S(e,i);D(d);N(e);S(v,d);N(o),

N(s);I(i);N(e);N(d);D(e);D(v);N(o).

Intuitively we can observe that the last sequence is the one more likely, because we know the phenomenon of diphthongization in Italian and the correspondance between endings -*evo* of imperfect and -*o* of present, but the choice of the correct sequence is left to inter-paradigmatic analysis.

3.5 IS IT WORTH THE COMPLEXITY?

The method outlined above for detecting invariance in word forms is computationally expensive, with a number of comparisons and choices (i.e., program steps) proportional to the product of the lengths of the two word forms processed. But it is useful in capturing common patterns of allomorphy, like radical vowel changes, palatalisation, and so on, and essential in processing both suffixal and prefixal morphology at the same time, and especially in processing discontinuous morphology, as found, for example, in semitic languages.

Here I give some short examples, for vowel change in German:

sprechen → spricht: N(spr);S(e,i);N(ch);S(en,t)

vowel alternation, suffixal and prefixal morphology, again in German:

spricht → gesprochen: I(ge);N(spr);S(i,o);N(ch);S(t,en)

A perfect third person masculine in arabic, *faʕala* ‘he did’, compared with the same paradigm cell for verb *ɣarra* ‘he seduced’, with gemination of second and third radicals, and *saʔima* ‘he hated’, with vocalization *-i-*:

faʕala → ɣarra: S(f,ɣ);N(a);S(ʕal,rr);N(a)

faʕala → saʔima: S(f,s);N(a);S(ʕal,ʔim);N(a)

From how the algorithm is built, and seen these and many other results, I am confident that this algorithm is more flexible, than the traditional approach of just aligning any end of the word to extract invariance.

3.6 SIMPLIFYING DATA: A QUICK SEARCH FOR INVARIANCE

The first step is finding invariance between forms of every lexeme for each single paradigm cell, and separate the invariant segments from the segments proper of each verbal form.

The database I prepared contains the following lexemes:

ABBANDONARE	‘to abandon’
ABOLIRE	‘to abolish’
ACCENDERE	‘to turn on, to light up‘
ADDURRE	‘to put forward’
AFFIGGERE	‘to post up’
AFFLIGGERE	‘to afflict’
AGGREDIRE	‘to assault’
ALLUDERE	‘to allude’
AMARE	‘to love’
ANDARE (overabundant)	‘to go’
APPARIRE	‘to appear’
APRIRE	‘to open’
ARDERE	‘to burn’
ASSOLVERE	‘to absolve’
ASSUMERE	‘to assume’
AVERE	‘to have’
BATTERE	‘to beat’
BERE	‘to drink’
CADERE	‘to fall’
CAMBIARE	‘to change’
CAPIRE	‘to understand’
CARPIRE	‘to wheedle’
CEDERE (overabundant)	‘to cede, to break down’
CHIEDERE	‘to ask’
CHIUDERE	‘to close’
CINGERE	‘to enclose’
COGLIERE	‘to pick’
COMPIERE	‘to accomplish’
CONCEDERE	‘to grant, to concede’
CONCERNERE (defective)	‘to concern’
CONDURRE	‘to drive, to direct, to conduct’
CONNETTERE	‘to connect’
CONOSCERE	‘to know’
CONSUMARE	‘to wear out’
CONTROLLARE	‘to control, to check’
CONVERTIRE	‘to convert’
COPIARE	‘to copy’
CORRERE	‘to run’
COSTRUIRE	‘to build’
CREDERE (overabundant)	‘to believe’
CREPARE	‘to crack’
CRESCERE	‘to grow’
CUOCERE (overabundant)	‘to cook’

DARE (overabundant)	‘to give’
DEVOLVERE (overabundant)	‘to devolve’
DIFENDERE	‘to defend’
DIPINGERE	‘to paint’
DIRE	‘to say’
DIRIGERE	‘to direct’
DISCUTERE	‘to discuss’
DISTINGUERE	‘to distinguish’
DISTRUGGERE	‘to destroy’
DOLERE	‘to ache’
DOMARE	‘to tame’
DOVERE (defective, overabundant)	‘to must’
EMETTERE	‘to emit’
ESIGERE	‘to require’
ESISTERE	‘to exist’
ESPELLERE	‘to expel’
ESPRIMERE	‘to express’
ESSERE	‘to be’
ESTINGUERE	‘to extinguish’
FARE (overabundant)	‘to do, to make’
FERIRE	‘to wound’
FONDERE	‘to melt’
FREGARE	‘to rub’
FRIGGERE	‘to fry’
FUGGIRE	‘to flee’
GEMERE (overabundant)	‘to moan’
GODERE	‘to enjoy’
INCUTERE	‘to instil’
INFRANGERE	‘to smash’
LEDERE	‘to hurt’
LEGGERE	‘to read’
MASTICARE	‘to chew’
MERITARE	‘to deserve’
METTERE	‘to put’
MORDERE	‘to bite’
MORIRE	‘to die’
MUOVERE (overabundant)	‘to move’
NASCERE	‘to be born’
NASCONDERE	‘to hide’
OSARE	‘to dare’
PARERE	‘to seem’
PASCERE (overabundant)	‘to graze’
PERCUOTERE	‘to hit’

PERDERE	‘to lose’
PIACERE	‘to be liked’
PORGERE	‘to hand out’
PORRE	‘to put’
POSSEDERE (overabundant)	‘to possess’
PREDILIGERE	‘to favour’
PRENDERE	‘to take’
PROVVEDERE	‘to provide for’
PUNGERE	‘to sting’
REDIGERE	‘to draft’
RESTRINGERE	‘to shrink’
RICEVERE (overabundant)	‘to receive’
RIFLETTERE	‘to reflect’
RIMANERE	‘to remain’
RISCUOTERE	‘to collect’
RISOLVERE	‘to resolve’
RISPONDERE	‘to reply’
RODERE	‘to gnaw’
ROMPERE	‘to break’
SALIRE	‘to rise, to climb’
SAPERE	‘to know’
SCEGLIERE	‘to choose’
SCENDERE	‘to descend’
SCRIVERE	‘to write’
SCUOTERE	‘to shake’
SEDERE (overabundant)	‘to sit’
SENTIRE	‘to hear’
SOLERE (defective)	‘to have the habit of’
SPARGERE	‘to scatter’
SPEGNERE	‘to turn off, to put out’
SPLENDERE (defective, overabundant)	‘to shine’
STARE (overabundant)	‘to stay’
STRIDERE (defective, overabundant)	‘to creak’
STRINGERE	‘to tighten’
TACERE	‘to keep quiet’
TEMERE (overabundant)	‘to fear’
TENDERE	‘to tend, to stretch’
TENERE	‘to keep’
TOGLIERE	‘to remove’
TORCERE	‘to twist’
TRARRE	‘to pull’
UDIRE	‘to hear’
USCIRE	‘to exit’

VALERE	‘to be worth, to mean’
VEDERE (overabundant)	‘to see’
VENIRE	‘to come’
VINCERE	‘to win’
VIVERE	‘to live’
VOLERE	‘to want’
VOLGERE	‘to direct, to turn’

All –and only– the synthetic forms of the paradigm are generated from the database, namely the forms of Indicative Present, Imperfect, Simple Past, and Future, Subjunctive Present and Imperfect, Conditional Present, Infinitive Present, Gerund Present, Participle Present and Past, and Imperative. Regarding Imperative, it is arguable what exactly are to be considered forms of Imperative; I follow Graffi (1996) in listing imperative form as 2S, 2P, 1P, the latter being idiosyncratic as used in a strictly inclusive sense; however, the First Plural Imperative form is not included in the database, being synchretic with First Plural Present Indicative and Subjunctive.

3.7 FINDING INVARIANCE FOR EACH PARADIGM CELL

This is accomplished running my implementation of the Levenshtein algorithm for each pair of forms, finding the largest pattern of invariance common to all forms.

For every lexeme L_n the program creates a list of forms F_{LnC} for the paradigm cell C. A list is needed because some verbs are overabundant, and a single result is not enough. Then, for every pair of lexemes $<L_n, L_{m>}$, every pair of forms $<f_1, f_2>$, f_1 from F_{LnC} and f_2 from F_{LmC}, are subjected to

the Levenshtein algorithm, which gives a result of a list of transformation paths (cf. § 3.4, page 51). The paths are encoded as models of invariance.

This scheme simulates how the program finds the model for indicative present, first person plural.

abbandonj'amo, abbj'amo → abb*j'amo	[andon] []
abbj'amo, abolj'amo → ab+j'amo	[b] [ol]
abbj'amo, kredj'amo → +j'amo	[abb] [kred]
kredj'amo, fattʃ'amo → +'amo	[kredj] [fattʃ]

The largest model common to all indicative present, first person plural forms is "+'amo". This process also selects segments, which in this context I will simply call "stems", that are to be applied to the model (that is, substituted for "+" and "*" symbols in the model) to generate the verbal form.

The following Table 11 shows the results of the computation:

paradigm cell	*model*
V.IND.PRES.1S	+
V.IND.PRES.2S	+
V.IND.PRES.3S	+
V.IND.PRES.1P	+'amo
V.IND.PRES.2P	+te
V.IND.PRES.3P	+no
V.IND.IMPF.1S	+o
V.IND.IMPF.2S	+i
V.IND.IMPF.3S	+a
V.IND.IMPF.1P	+v'amo
V.IND.IMPF.2P	+v'ate
V.IND.IMPF.3P	+ano
V.IND.PREM.1S	+
V.IND.PREM.2S	+sti
V.IND.PREM.3S	+
V.IND.PREM.1P	+mmo
V.IND.PREM.2P	+ste
V.IND.PREM.3P	+o
V.IND.FUT.1S	+r'ɔ
V.IND.FUT.2S	+r'aj
V.IND.FUT.3S	+r'a
V.IND.FUT.1P	+r'emo
V.IND.FUT.2P	+r'ete
V.IND.FUT.3P	+r'anno
V.CONG.PRES.1S	+
V.CONG.PRES.2S	+
V.CONG.PRES.3S	+
V.CONG.PRES.1P	+'amo
V.CONG.PRES.2P	+'ate

paradigm cell	*model*
V.CONG.PRES.3P	+no
V.CONG.IMPF.1S	+ssi
V.CONG.IMPF.2S	+ssi
V.CONG.IMPF.3S	+sse
V.CONG.IMPF.1P	+ssimo
V.CONG.IMPF.2P	+ste
V.CONG.IMPF.3P	+ssero
V.COND.PRES.1S	+r'ɛj
V.COND.PRES.2S	+r'esti
V.COND.PRES.3S	+r'ɛbbe
V.COND.PRES.1P	+r'emmo
V.COND.PRES.2P	+r'este
V.COND.PRES.3P	+r'ɛbbero
V.IMP.2S	+
V.IMP.2P	+te
V.GERU	+ndo
V.INF	+re
V.PART.PRES.S	+nte
V.PART.PRES.P	+nti
V.PART.PASS.M.S	+o
V.PART.PASS.M.P	+i
V.PART.PASS.F.S	+a
V.PART.PASS.F.P	+e

Table 11: invariant parts of Italian verbal paradigm cells.

3.8 REDUCING CELL MODELS

On a successive step, the set C of paradigm cells c_i is partitioned so that every lexeme has the same stem (or the same set of stems) for the model computed for those cells.

As shown above, every lexeme has a stem for every model, that is, for every paradigm cell.

Pairs of paradigm cells are compared to see if every lexeme happens to have the same stems for the two cells. Where this is the case, the two cells are grouped. This process is iterated for every paradigm cell, so that the entire paradigm is partitioned in metacells.

For instance, it is immediately clear that the four forms corresponding to masculine singular, masculine plural, feminine singular and feminine plural of past participle, in Italian, differ only by the ending vowel. And that vowel is, respectively, the same for every lexeme; these ending are not dictated on conjugation classes. Consequence of this fact is that every lexeme has the same stem for those four paradigm cells.

cell	*model*	*stems*		
		AMARE	CREDERE	CORRERE
V.PART.PASS.M.S	+o	am'at	kred'ut	k'ors
V.PART.PASS.M.P	+i	am'at	kred'ut	k'ors
V.PART.PASS.F.S	+a	am'at	kred'ut	k'ors
V.PART.PASS.F.P	+e	am'at	kred'ut	k'ors

Table 12: grouping cells of PAST PARTICIPLE where stems are systematically identical.

Thus, the program will group these cells together in one metacell. The same happens for all future indicative and present conditional forms:

cell	*model*	*stems*			
		AMARE	BERE	CADERE	CAPIRE
V.FUT.PRES.1S	+r'ɔ	ame	ber	kad	kapi
V.COND.PRES.1S	+r'ɛj	ame	ber	kad	kapi
(etc.)		ame	ber	kad	kapi

Table 13: grouping cells of FUTURE and CONDITIONAL where stems are systematically identical

And the program will group those six paradigm cells into another metacell.

The list of metacells and ungrouped cells, contains 21 elements, reducing the 51 forms of the whole paradigm:

1. V.IND.PRES.1S
2. V.IND.PRES.2S
3. V.IND.PRES.3S
4. V.MC_4
    - V.IND.PRES.1P
    - V.CONG.PRES.1P
    - V.CONG.PRES.2P
5. V.IND.PRES.2P
6. V.IND.PRES.3P
7. V.MC_7
    - V.IND.IMPF.1S
    - V.IND.IMPF.2S
    - V.IND.IMPF.3S
    - V.IND.IMPF.3P
8. V.MC_8
    - V.IND.IMPF.1P
    - V.IND.IMPF.2P
9. V.IND.PREM.1S
10. V.MC_10
    - V.IND.PREM.2S
    - V.IND.PREM.2P
    - V.CONG.IMPF.1S
    - V.CONG.IMPF.2S
    - V.CONG.IMPF.3S
    - V.CONG.IMPF.1P
    - V.CONG.IMPF.2P
    - V.CONG.IMPF.3P
11. V.IND.PREM.3S
12. V.IND.PREM.1P
13. V.IND.PREM.3P
14. V.MC_14
    - V.IND.FUT.1S

V.IND.FUT.2S
V.IND.FUT.3S
V.IND.FUT.1P
V.IND.FUT.2P
V.IND.FUT.3P
V.COND.PRES.1S
V.COND.PRES.2S
V.COND.PRES.3S
V.COND.PRES.1P
V.COND.PRES.2P
V.COND.PRES.3P

15. V.MC_15
    V.CONG.PRES.1S
    V.CONG.PRES.2S
    V.CONG.PRES.3S
    V.CONG.PRES.3P
16. V.IMP.2S
17. V.IMP.2P
18. V.GERU
19. V.INF
20. V.MC_20
    V.PART.PRES.S
    V.PART.PRES.P
21. V.MC_21
    V.PART.PASS.M.S
    V.PART.PASS.M.P
    V.PART.PASS.F.S
    V.PART.PASS.F.P

3.9 SIMILARITY BETWEEN METACELLS

The following table shows the number of lexemes where the stems for the two given metacells are the same.

	V.MC_21	V.MC_20	V.INF	V.GERU	V.IMP.2P	V.IMP.2S	V.MC_15	V.MC_14	V.IND.PREM.3P	V.IND.PREM.1P	V.IND.PREM.3S	V.MC_10	V.IND.PREM.1S	V.MC_8	V.MC_7	V.IND.PRES.3P	V.IND.PRES.2P	V.MC_4	V.IND.PRES.3S	V.IND.PRES.2S
V.IND.PRES.1S	3	1	0	0	1	1	0	1	0	0	0	0	0	0	0	117	0	0	0	0
V.IND.PRES.2S	3	1	0	0	1	115	12	1	0	0	0	0	0	0	0	0	0	0	0	
V.IND.PRES.3S	3	3	78	2	4	13	0	1	0	0	0	0	0	0	0	12	3	0		
V.MC_4	3	1	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0			
V.IND.PRES.2P	3	115	53	116	132	2	0	1	0	131	23	131	0	0	0	0				
V.IND.PRES.3P	3	1	0	0	1	13	0	1	0	0	0	0	0	0	0					
V.MC_7	3	1	0	0	1	1	0	1	0	0	0	0	0	0						
V.MC_8	3	1	0	0	1	1	0	101	0	0	0	0	0							
V.IND.PREM.1S	3	1	0	0	1	1	0	1	0	0	0	0								
V.MC_10	3	115	49	116	128	1	0	1	0	135	23									
V.IND.PREM.3S	3	9	17	8	24	1	0	1	0	24										
V.IND.PREM.1P	3	115	49	116	128	1	0	1	0											
V.IND.PREM.3P	3	1	0	0	1	1	0	1												
V.MC_14	4	2	1	1	2	2	1													
V.MC_15	3	1	0	0	1	1														
V.IMP.2S	4	1	2	1	2															
V.IMP.2P	4	113	50	113																
V.GERU	3	135	35																	
V.INF	3	34																		
V.MC_20	4																			

Table 14: number of lexemes for which two stems are identical.

Cells with higher numbers suggest that few highly irregular verbs block the grouping between the paradigm cell on the row and the paradigm cell on the column. These correspond broadly to the cases of positive but low values of entropy in Bonami and Boyé's analysis (see page 43).

3.10 FINDING RELATIONS BETWEEN LEXEME STEMS

The next step of the computation, after calculating every lexeme stem for every metacell, consists in finding out the formal relations between these stems.

The implementation of Levenshtein's algorithm calculates the difference between each couple of stems, intraparadigmatically for every lexeme. Then, the lexemes whose inflected forms obey to the same relation between two given stems, are grouped. That is as saying that, for any given couple of metacells, every relation is associated with a set of lexemes.

The purpose of this computation, is to see how –and how predictably– lexemes form "classes", not conjugation classes in the traditional sense, but in relation to these relations.

The listing below is a small extract: the full listing consists of about 24.000 lines[7].

V.IND.PRES.1S → V.IND.PRES.2S

N;S(o,i) (72):	ABBANDONARE, ACCENDERE, ALLUDERE, AMARE, APRIRE, ARDERE, ASSOLVERE, ASSUMERE, BATTERE, BERE, CADERE, CEDERE, CHIEDERE, CHIUDERE, CONCEDERE, CONCERNERE, CONNETTERE, CONSUMARE, CONTROLLARE, CONVERTIRE, CORRERE, CREDERE, CREPARE, CUOCERE, DEVOLVERE, DIFENDERE, DISCUTERE, DISTINGUERE, DOMARE, DOVERE, EMETTERE, ESISTERE, ESPELLERE, ESPRIMERE, ESTINGUERE, FONDERE, FREGARE, GEMERE, GODERE, INCUTERE, LEDERE, MASTICARE, MERITARE, METTERE, MORDERE, MUOVERE, NASCONDERE, OSARE, PERCUOTERE, PERDERE,

[7] The entire computation results can be downloaded at the URL: http://www.pascoli.it/thesis/thesis_files.zip

	POSSEDERE, PRENDERE, PROVVEDERE, RICEVERE, RIFLETTERE, RISCUOTERE, RISOLVERE, RISPONDERE, RODERE, ROMPERE, SCENDERE, SCRIVERE, SCUOTERE, SEDERE, SENTIRE, SPLENDERE, STRIDERE, TEMERE, TENDERE, UDIRE, VEDERE, VIVERE
N;S(skoˌʃʃi) (11):	ABOLIRE, AGGREDIRE, CAPIRE, CARPIRE, CONOSCERE, COSTRUIRE, CRESCERE, FERIRE, NASCERE, PASCERE, USCIRE
N;S(go,dʒi) (8):	CINGERE, DIRIGERE, ESIGERE, PORGERE, PREDILIGERE, REDIGERE, SPARGERE, VOLGERE
N;S(ggo,ddʒi) (6):	AFFIGGERE, AFFLIGGERE, DISTRUGGERE, FRIGGERE, FUGGIRE, LEGGERE
N;S(ŋgo,ndʒi) (5):	DIPINGERE, INFRANGERE, PUNGERE, RESTRINGERE, STRINGERE
N;S(ɔ,aj) (4):	AVERE, DARE, SAPERE, STARE
N;S(ko,tʃi) (4):	ADDURRE, CONDURRE, DIRE, TORCERE
N;S(jo,i) (3):	CAMBIARE, COMPIERE, COPIARE
N;S(jo,ri) (3):	APPARIRE, MORIRE, PARERE
N;S(lgo,ʎʎi) (3):	COGLIERE, SCEGLIERE, TOGLIERE
N;S(go,i) (2):	SALIRE, VALERE
N;S(ŋgo,ni) (2):	PORRE, RIMANERE
N;D(t);N;S(o,i) (2):	PIACERE, TACERE
N;I(j);N;S(ggo,di) (2):	POSSEDERE, SEDERE
N;I(j);N;S(ŋgo,ni) (2):	TENERE, VENIRE
N;S(do,j) (1):	ANDARE
N;S(ggo,i) (1):	TRARRE
N;S(bbo,vi) (1):	DOVERE
N;S(ono,ɛj) (1):	ESSERE
N;S(ttʃo,j) (1):	FARE
N;S(ŋgo,ɲɲi) (1):	SPEGNERE
N;S(ŋko,ntʃi) (1):	VINCERE
N;S(eŋgo,jeni) (1):	TENERE
N;I(w);N;S(go,i) (1):	DOLERE
N;I(w);N;S(ʎʎo,i) (1):	VOLERE
N;I(w);N;S(ʎʎo,li) (1):	SOLERE

V.IND.PRES.1S → V.IND.PRES.3S

N;S(o,e) (63):	ACCENDERE, ALLUDERE, APRIRE, ARDERE, ASSOLVERE, ASSUMERE, BATTERE, BERE, CADERE, CEDERE, CHIEDERE, CHIUDERE, COMPIERE, CONCEDERE, CONCERNERE, CONNETTERE, CONVERTIRE, CORRERE, CREDERE, CUOCERE, DEVOLVERE, DIFENDERE, DISCUTERE, DISTINGUERE, DOVERE, EMETTERE, ESISTERE, ESPELLERE, ESPRIMERE, ESTINGUERE, FONDERE, GEMERE, GODERE, INCUTERE, LEDERE, METTERE, MORDERE, MUOVERE, NASCONDERE, PERCUOTERE, PERDERE, POSSEDERE, PRENDERE, PROVVEDERE, RICEVERE, RIFLETTERE, RISCUOTERE, RISOLVERE, RISPONDERE, RODERE, ROMPERE, SCENDERE, SCRIVERE, SCUOTERE, SEDERE, SENTIRE, SPLENDERE,

	STRIDERE, TEMERE, TENDERE, UDIRE, VEDERE, VIVERE
N;S(o,a) (12):	ABBANDONARE, AMARE, CAMBIARE, CONSUMARE, CONTROLLARE, COPIARE, CREPARE, DOMARE, FREGARE, MASTICARE, MERITARE, OSARE
N;S(sko,ʃʃe) (11):	ABOLIRE, AGGREDIRE, CAPIRE, CARPIRE, CONOSCERE, COSTRUIRE, CRESCERE, FERIRE, NASCERE, PASCERE, USCIRE
N;S(go,dʒe) (8):	CINGERE, DIRIGERE, ESIGERE, PORGERE, PREDILIGERE, REDIGERE, SPARGERE, VOLGERE
N;S(ggo,ddʒe) (6):	AFFIGGERE, AFFLIGGERE, DISTRUGGERE, FRIGGERE, FUGGIRE, LEGGERE
N;S(ŋgo,ndʒe) (5):	DIPINGERE, INFRANGERE, PUNGERE, RESTRINGERE, STRINGERE
N;S(ɔ,a) (4):	AVERE, DARE, SAPERE, STARE
N;S(ko,tʃe) (4):	ADDURRE, CONDURRE, DIRE, TORCERE
N;S(jo,re) (3):	APPARIRE, MORIRE, PARERE
N;S(lgo,ʎʎe) (3):	COGLIERE, SCEGLIERE, TOGLIERE
N;S(go,e) (2):	SALIRE, VALERE
N;S(ŋgo,ne) (2):	PORRE, RIMANERE
N;D(t);N;S(o,e) (2):	PIACERE, TACERE
N;I(j);N;S(ggo,de) (2):	POSSEDERE, SEDERE
N;I(j);N;S(ŋgo,ne) (2):	TENERE, VENIRE
N;I(w);N;S(ʎʎo,le) (2):	SOLERE, VOLERE
N;D(do) (1):	ANDARE
N;D(ttʃo) (1):	FARE
N;S(ggo,e) (1):	TRARRE
N;S(bbo,ve) (1):	DOVERE
N;S(ŋgo,ɲɲe) (1):	SPEGNERE
N;S(ŋko,ntʃe) (1):	VINCERE
N;S(eŋgo,jene) (1):	TENERE
N;I(w);N;S(go,e) (1):	DOLERE
N;D(s);N;S(ono,ɛ) (1):	ESSERE

V.IND.PRES.3S → V.IND.PREM.3S

N;D(');N;I(') (12):	BATTERE, CEDERE, CONNETTERE, CREDERE, ESIGERE, ESISTERE, GEMERE, PASCERE, RICEVERE, SPLENDERE, STRIDERE, TEMERE
N;S(dʒ,s) (9):	CINGERE, DIPINGERE, INFRANGERE, PORGERE, PUNGERE, RESTRINGERE, SPARGERE, STRINGERE, VOLGERE
N;D(');N;S(a,'ɔ) (7):	ABBANDONARE, AMARE, CAMBIARE, CONSUMARE, CREPARE, FREGARE, MASTICARE
N;D(ʃʃe) (6):	ABOLIRE, AGGREDIRE, CAPIRE, CARPIRE, COSTRUIRE, FERIRE
N;S(nd,z) (6):	ACCENDERE, DIFENDERE, NASCONDERE, RISPONDERE, SCENDERE, TENDERE
N;S(ddʒ,ss) (5):	AFFIGGERE, AFFLIGGERE, DISTRUGGERE, FRIGGERE, LEGGERE
N;S(d,z) (4):	ALLUDERE, CHIEDERE, CHIUDERE, LEDERE
N;D(');N;S(e,'i) (4):	APRIRE, FUGGIRE, SALIRE, SENTIRE
N;S('ɔ,o);N;S(a,'ɔ) (4):	CONTROLLARE, COPIARE, DOMARE, OSARE
N;I(v) (3):	APPARIRE, BERE, PARERE

N;S(d,s) (3):	ARDERE, MORDERE, PERDERE
N;S(tʃ,ss) (3):	ADDURRE, CONDURRE, DIRE
N;S(ʎʎ,ls) (3):	COGLIERE, SCEGLIERE, TOGLIERE
N;D(');N;I('ett) (3):	PASCERE, SPLENDERE, STRIDERE
N;D(w);N;S(t,ss) (3):	PERCUOTERE, RISCUOTERE, SCUOTERE
N;D(');N;I(');N;I(tte) (3):	PASCERE, SPLENDERE, STRIDERE
N;S(e,i) (2):	PROVVEDERE, VEDERE
N;S(n,z) (2):	PORRE, RIMANERE
N;S(v,s) (2):	ASSOLVERE, RISOLVERE
N;S(t,ss) (2):	DISCUTERE, INCUTERE
N;S(tʃ,s) (2):	TORCERE, VINCERE
N;S(v,ss) (2):	SCRIVERE, VIVERE
N;S(ʃʃ,bb) (2):	CONOSCERE, CRESCERE
N;S(ett,iz) (2):	EMETTERE, METTERE
N;S(tʃ,kkw) (2):	PIACERE, TACERE
N;S(ŋgw,ns) (2):	DISTINGUERE, ESTINGUERE
N;S(idʒ,ɛss) (2):	DIRIGERE, PREDILIGERE
N;D(j);N;I(n) (2):	TENERE, VENIRE
N;S(j'ɛd,ed') (2):	POSSEDERE, SEDERE
N;S('ɔ,o);N;I(') (2):	DEVOLVERE, GODERE
N;S('ɛ,e);N;I(') (2):	CONCERNERE, RIFLETTERE
N;I(ed);N;S(d,tt) (2):	CEDERE, CREDERE
N;I(em);N;S(m,tt) (2):	GEMERE, TEMERE
N;I(j);N;S(a,ede) (2):	DARE, STARE
N;S(j'ɛ,e);N;I(') (2):	POSSEDERE, SEDERE
N;S(j,ed);N;D(ɛd) (2):	POSSEDERE, SEDERE
N;S(j,ed);N;S(ɛd,ett) (2):	POSSEDERE, SEDERE
N;I(d) (1):	CADERE
N;I(s) (1):	VALERE
N;I(ss) (1):	TRARRE
N;S(r,s) (1):	CORRERE
N;S(m,ns) (1):	ASSUMERE
N;S(je,en) (1):	TENERE
N;S(om,up) (1):	ROMPERE
N;S(ɔd,oz) (1):	RODERE
N;S(ɲɲ,ns) (1):	SPEGNERE
N;S(a,ebbe) (1):	AVERE
N;S(a,eppe) (1):	SAPERE
N;S(a,etʃe) (1):	FARE
N;S(ed,ɛss) (1):	CONCEDERE
N;S(im,ɛss) (1):	ESPRIMERE
N;S(ond,uz) (1):	FONDERE
N;S(ɛnd,ez) (1):	PRENDERE
N;S(ʃʃ,kkw) (1):	NASCERE
N;S('ev,ov') (1):	DOVERE
N;S('ɔd,od') (1):	GODERE
N;S(idʒ,ass) (1):	REDIGERE
N;D(w);N;I(l) (1):	VOLERE
N;S(w'ɔl,ol') (1):	SOLERE
N;S('ɔde,ud'i) (1):	UDIRE
N;S('ɔma,om'ɔ) (1):	DOMARE
N;S('ɔza,oz'ɔ) (1):	OSARE
N;I(f);N;S(ɛ,u) (1):	ESSERE
N;I(om);N;D(ma) (1):	DOMARE
N;I(oz);N;D(za) (1):	OSARE
N;S(w'ɔre,or'i) (1):	MORIRE
N;D(w);N;S(v,ss) (1):	MUOVERE

N;S('e,o);N;I(') (1):	DOVERE
N;D(');N;S(je,'i) (1):	COMPIERE
N;D(w);N;S(tʃ,ss) (1):	CUOCERE
N;I(ev);N;S(v,tt) (1):	RICEVERE
N;I(ov);N;S(v,tt) (1):	DOVERE
N;S(w'ɔ,o);N;I(') (1):	SOLERE
N;S(w,ol);N;D(ɔl) (1):	SOLERE
N;S(ɛ,u);N;S(l,s) (1):	ESPELLERE
N;I(end);N;S(nd,tt) (1):	SPLENDERE
N;I(id);N;S(id,ett) (1):	STRIDERE
N;S('e,u);N;S(e,'i) (1):	USCIRE
N;S('ɔ,o);N;I('ett) (1):	DEVOLVERE
N;S('ɔ,u);N;S(e,'i) (1):	UDIRE
N;S('ɛ,e);N;S(a,'ɔ) (1):	MERITARE
N;S('ɛ,e);N;S(e,'i) (1):	CONVERTIRE
N;S('ɛ,et);N;S(t,') (1):	RIFLETTERE
N;S(v,and);N;S(a,ɔ) (1):	ANDARE
N;S(w'ɔ,o);N;S(e,'i) (1):	MORIRE
N;S(w,or);N;S(ɔre,i) (1):	MORIRE
N;I(olv);N;S(ɔlv,ett) (1):	DEVOLVERE
N;I(vol);N;S(ɔlv,ett) (1):	DEVOLVERE
N;S('e,uʃ);N;S(ʃe,'i) (1):	USCIRE
N;S('ɔ,ol);N;S(la,'ɔ) (1):	CONTROLLARE
N;D(w);N;S(ɔ,o);N;I(s) (1):	DOLERE
N;S('ɔ,o);N;I(');N;I(tte) (1):	DEVOLVERE

V.IND.PRES.3S → V.MC_15 (V.CONG.PRES.1S)

N;S(e,a) (63):	ACCENDERE, ALLUDERE, APRIRE, ARDERE, ASSOLVERE, ASSUMERE, BATTERE, BERE, CADERE, CEDERE, CHIEDERE, CHIUDERE, COMPIERE, CONCEDERE, CONCERNERE, CONNETTERE, CONVERTIRE, CORRERE, CREDERE, CUOCERE, DEVOLVERE, DIFENDERE, DISCUTERE, DISTINGUERE, DOVERE, EMETTERE, ESISTERE, ESPELLERE, ESPRIMERE, ESTINGUERE, FONDERE, GEMERE, GODERE, INCUTERE, LEDERE, METTERE, MORDERE, MUOVERE, NASCONDERE, PERCUOTERE, PERDERE, POSSEDERE, PRENDERE, PROVVEDERE, RICEVERE, RIFLETTERE, RISCUOTERE, RISOLVERE, RISPONDERE, RODERE, ROMPERE, SCENDERE, SCRIVERE, SCUOTERE, SEDERE, SENTIRE, SPLENDERE, STRIDERE, TEMERE, TENDERE, UDIRE, VEDERE, VIVERE
N;S(ʃʃe,ska) (11):	ABOLIRE, AGGREDIRE, CAPIRE, CARPIRE, CONOSCERE, COSTRUIRE, CRESCERE, FERIRE, NASCERE, PASCERE, USCIRE
N;S(a,i) (10):	ABBANDONARE, AMARE, CONSUMARE, CONTROLLARE, CREPARE, DOMARE, FREGARE, MASTICARE, MERITARE, OSARE
N;S(dʒe,ga) (8):	CINGERE, DIRIGERE, ESIGERE, PORGERE, PREDILIGERE, REDIGERE, SPARGERE, VOLGERE

N;S(ddʒe,gga) (6):	AFFIGGERE, AFFLIGGERE, DISTRUGGERE, FRIGGERE, FUGGIRE, LEGGERE
N;S(ndʒe,ŋga) (5):	DIPINGERE, INFRANGERE, PUNGERE, RESTRINGERE, STRINGERE
N;S(tʃe,ka) (4):	ADDURRE, CONDURRE, DIRE, TORCERE
N;S(re,ja) (3):	APPARIRE, MORIRE, PARERE
N;S(ʎʎe,lga) (3):	COGLIERE, SCEGLIERE, TOGLIERE
N;I(i) (2):	DARE, STARE
N;S(e,ga) (2):	SALIRE, VALERE
N;S(ja,i) (2):	CAMBIARE, COPIARE
N;S(ne,ŋga) (2):	PORRE, RIMANERE
N;I(t);N;S(e,a) (2):	PIACERE, TACERE
N;D(j);N;S(de,gga) (2):	POSSEDERE, SEDERE
N;D(j);N;S(ne,ŋga) (2):	TENERE, VENIRE
N;D(w);N;S(le,ʎʎa) (2):	SOLERE, VOLERE
N;I(ad) (1):	ANDARE
N;I(da) (1):	ANDARE
N;I(abbj) (1):	AVERE
N;I(appj) (1):	SAPERE
N;I(attʃ) (1):	FARE
N;I(bbja) (1):	AVERE
N;I(ppja) (1):	SAPERE
N;I(ttʃa) (1):	FARE
N;S(e,gga) (1):	TRARRE
N;S(ve,bba) (1):	DOVERE
N;S(ŋŋe,ŋga) (1):	SPEGNERE
N;S(ntʃe,ŋka) (1):	VINCERE
N;S(jene,eŋga) (1):	TENERE
N;D(w);N;S(e,ga) (1):	DOLERE
N;I(s);N;S(ɛ,ia) (1):	ESSERE

V.IND.PRES.3S → V.IMP.2S

N;S(e,i) (113):	ABOLIRE, ACCENDERE, ADDURRE, AFFIGGERE, AFFLIGGERE, AGGREDIRE, ALLUDERE, APPARIRE, APRIRE, ARDERE, ASSOLVERE, ASSUMERE, BATTERE, BERE, CADERE, CAPIRE, CARPIRE, CEDERE, CHIEDERE, CHIUDERE, CINGERE, COGLIERE, CONCEDERE, CONCERNERE, CONDURRE, CONNETTERE, CONOSCERE, CONVERTIRE, CORRERE, COSTRUIRE, CREDERE, CRESCERE, CUOCERE, DEVOLVERE, DIFENDERE, DIPINGERE, DIRIGERE, DISCUTERE, DISTINGUERE, DISTRUGGERE, DOLERE, EMETTERE, ESIGERE, ESISTERE, ESPELLERE, ESPRIMERE, ESTINGUERE, FERIRE, FONDERE, FRIGGERE, FUGGIRE, GEMERE, GODERE, INCUTERE, INFRANGERE, LEDERE, LEGGERE, METTERE, MORDERE, MORIRE, MUOVERE, NASCERE, NASCONDERE, PARERE, PASCERE, PERCUOTERE, PERDERE, PIACERE, PORGERE, PORRE, POSSEDERE, PREDILIGERE, PRENDERE, PROVVEDERE, PUNGERE, REDIGERE, RESTRINGERE, RICEVERE, RIFLETTERE, RIMANERE, RISCUOTERE, RISOLVERE,

	RISPONDERE, RODERE, ROMPERE, SALIRE, SCEGLIERE, SCENDERE, SCRIVERE, SCUOTERE, SEDERE, SENTIRE, SOLERE, SPARGERE, SPEGNERE, SPLENDERE, STRIDERE, STRINGERE, TACERE, TEMERE, TENDERE, TENERE, TOGLIERE, TORCERE, TRARRE, UDIRE, USCIRE, VALERE, VEDERE, VENIRE, VINCERE, VIVERE, VOLGERE
N (16):	ABBANDONARE, AMARE, ANDARE, CAMBIARE, CONSUMARE, CONTROLLARE, COPIARE, CREPARE, DARE, DOMARE, FARE, FREGARE, MASTICARE, MERITARE, OSARE, STARE
N;I(j) (4):	ANDARE, DARE, FARE, STARE
N;D(tʃe) (1):	DIRE
N;I(bbi) (1):	AVERE
N;I(ppi) (1):	SAPERE
N;S(je,i) (1):	COMPIERE
N;I(s);N;S(ɛ,ij) (1):	ESSERE
N;D(w);N;S(le,ʎʎi) (1):	VOLERE

V.MC_4 (V.IND.PRES.1P) → V.MC_7 (V.IND.IMPF.1S)

N;S(j,'ev) (62):	ACCENDERE, ALLUDERE, ARDERE, ASSOLVERE, ASSUMERE, BATTERE, BERE, CADERE, CEDERE, CHIEDERE, CHIUDERE, CONCEDERE, CONCERNERE, CONNETTERE, CORRERE, CREDERE, DEVOLVERE, DIFENDERE, DISCUTERE, DISTINGUERE, DOLERE, EMETTERE, ESISTERE, ESPELLERE, ESPRIMERE, ESTINGUERE, FONDERE, GEMERE, GODERE, INCUTERE, LEDERE, METTERE, MORDERE, MUOVERE, NASCONDERE, PERCUOTERE, PERDERE, PORRE, POSSEDERE, PRENDERE, PROVVEDERE, RICEVERE, RIFLETTERE, RIMANERE, RISCUOTERE, RISOLVERE, RISPONDERE, RODERE, ROMPERE, SCENDERE, SCRIVERE, SCUOTERE, SEDERE, SPLENDERE, STRIDERE, TEMERE, TENDERE, TENERE, TRARRE, VALERE, VEDERE, VIVERE
N;I('ev) (34):	ADDURRE, AFFIGGERE, AFFLIGGERE, CINGERE, COGLIERE, CONDURRE, CONOSCERE, CRESCERE, CUOCERE, DIPINGERE, DIRE, DIRIGERE, DISTRUGGERE, ESIGERE, FRIGGERE, INFRANGERE, LEGGERE, NASCERE, PASCERE, PIACERE, PORGERE, PREDILIGERE, PUNGERE, REDIGERE, RESTRINGERE, SCEGLIERE, SPARGERE, SPEGNERE, STRINGERE, TACERE, TOGLIERE, TORCERE, VINCERE, VOLGERE
N;S(j,'iv) (15):	ABOLIRE, AGGREDIRE, APPARIRE, APRIRE, CAPIRE, CARPIRE, COMPIERE, CONVERTIRE, COSTRUIRE, FERIRE, MORIRE, SALIRE, SENTIRE, UDIRE, VENIRE
N;S(j,'av) (13):	ABBANDONARE, AMARE, ANDARE, CONSUMARE, CONTROLLARE, CREPARE, DARE, DOMARE, FREGARE, MASTICARE, MERITARE, OSARE, STARE

N;I('av) (2):	CAMBIARE, COPIARE
N;I('iv) (2):	FUGGIRE, USCIRE
N;S(ʎʎ,l'ev) (2):	SOLERE, VOLERE
N;S(bbj,v'ev) (2):	AVERE, DOVERE
N;S(j,r'ev) (1):	PARERE
N;S(pj,'ev) (1):	SAPERE
N;S(sj,'ɛr) (1):	ESSERE
N;S(tʃ,ʃ'ev) (1):	FARE
N;D(t);N;I('ev) (1):	FARE
N;D(w);N;I('ev) (1):	CUOCERE
N;I(w);N;I('ev) (1):	CUOCERE
N;D(w);N;S(j,'ev) (1):	MUOVERE
N;I(w);N;S(j,'ev) (1):	MUOVERE

V.IND.PRES.2P → V.MC_7 (V.IND.IMPF.1S)

N;I(v) (133):	ABBANDONARE, ABOLIRE, ACCENDERE, ADDURRE, AFFIGGERE, AFFLIGGERE, AGGREDIRE, ALLUDERE, AMARE, ANDARE, APPARIRE, APRIRE, ARDERE, ASSOLVERE, ASSUMERE, AVERE, BATTERE, BERE, CADERE, CAMBIARE, CAPIRE, CARPIRE, CEDERE, CHIEDERE, CHIUDERE, CINGERE, COGLIERE, COMPIERE, CONCEDERE, CONCERNERE, CONDURRE, CONNETTERE, CONOSCERE, CONSUMARE, CONTROLLARE, CONVERTIRE, COPIARE, CORRERE, COSTRUIRE, CREDERE, CREPARE, CRESCERE, CUOCERE, DARE, DEVOLVERE, DIFENDERE, DIPINGERE, DIRIGERE, DISCUTERE, DISTINGUERE, DISTRUGGERE, DOLERE, DOMARE, DOVERE, EMETTERE, ESIGERE, ESISTERE, ESPELLERE, ESPRIMERE, ESTINGUERE, FERIRE, FONDERE, FREGARE, FRIGGERE, FUGGIRE, GEMERE, GODERE, INCUTERE, INFRANGERE, LEDERE, LEGGERE, MASTICARE, MERITARE, METTERE, MORDERE, MORIRE, MUOVERE, NASCERE, NASCONDERE, OSARE, PARERE, PASCERE, PERCUOTERE, PERDERE, PIACERE, PORGERE, PORRE, POSSEDERE, PREDILIGERE, PRENDERE, PROVVEDERE, PUNGERE, REDIGERE, RESTRINGERE, RICEVERE, RIFLETTERE, RIMANERE, RISCUOTERE, RISOLVERE, RISPONDERE, RODERE, ROMPERE, SALIRE, SAPERE, SCEGLIERE, SCENDERE, SCRIVERE, SCUOTERE, SEDERE, SENTIRE, SOLERE, SPARGERE, SPEGNERE, SPLENDERE, STARE, STRIDERE, STRINGERE, TACERE, TEMERE, TENDERE, TENERE, TOGLIERE, TORCERE, TRARRE, UDIRE, USCIRE, VALERE, VEDERE, VENIRE, VINCERE, VIVERE, VOLERE, VOLGERE
N;D(w);N;I(v) (2):	CUOCERE, MUOVERE
N;I(w);N;I(v) (2):	CUOCERE, MUOVERE
N;D(sj);N;I(r) (1):	ESSERE
N;I(atʃ);N;S(a,ev) (1):	FARE
N;I(itʃ);N;S(i,ev) (1):	DIRE

V.MC_7 (V.IND.IMPF.1S) → V.MC_21 (V.PART.PASS.M.S)

N;S(v,t) (27):	ABBANDONARE, ABOLIRE, AGGREDIRE, AMARE, ANDARE, CAMBIARE, CAPIRE, CARPIRE, CONSUMARE, CONTROLLARE, CONVERTIRE, COPIARE, COSTRUIRE, CREPARE, DARE, DOMARE, FERIRE, FREGARE, FUGGIRE, MASTICARE, MERITARE, OSARE, SALIRE, SENTIRE, STARE, UDIRE, USCIRE
N;S(ev,ut) (26):	AVERE, BATTERE, BERE, CADERE, CEDERE, CONOSCERE, CREDERE, CRESCERE, DOLERE, DOVERE, GEMERE, GODERE, PASCERE, PERDERE, PIACERE, POSSEDERE, PROVVEDERE, RICEVERE, RIFLETTERE, SAPERE, SEDERE, TACERE, TEMERE, TENERE, VEDERE, VOLERE
N;D(end);N;S(v,z) (5):	ACCENDERE, DIFENDERE, PRENDERE, SCENDERE, TENDERE
N;I(');N;S(dʒ'ev,t) (4):	CINGERE, DIPINGERE, INFRANGERE, PUNGERE
N;S(wot'ev,'ɔss) (3):	PERCUOTERE, RISCUOTERE, SCUOTERE
N;D(ett);N;S(v,ss) (3):	CONNETTERE, EMETTERE, METTERE
N;D(wot);N;S(ev,ɔss) (3):	PERCUOTERE, RISCUOTERE, SCUOTERE
N;I(');N;S(ddʒ'ev,tt) (3):	AFFLIGGERE, DISTRUGGERE, FRIGGERE
N;S(ut'ev,'uss) (2):	DISCUTERE, INCUTERE
N;S(idʒ'ev,'att) (2):	ESIGERE, REDIGERE
N;S(idʒ'ev,'ɛtt) (2):	DIRIGERE, PREDILIGERE
N;S(olv'ev,'ɔlt) (2):	ASSOLVERE, RISOLVERE
N;S(ond'ev,'ɔst) (2):	NASCONDERE, RISPONDERE
N;S(oʎʎ'ev,'ɔlt) (2):	COGLIERE, TOGLIERE
N;S(utʃ'ev,'ott) (2):	ADDURRE, CONDURRE
N;S(iddʒ'ev,'itt) (2):	AFFLIGGERE, FRIGGERE
N;S(iŋgw'ev,'int) (2):	DISTINGUERE, ESTINGUERE
N;D(ud);N;S(ev,uz) (2):	ALLUDERE, CHIUDERE
N;I(');N;S(ʃ'ev,t) (2):	CUOCERE, FARE
N;D(ut);N;S(ev,uss) (2):	DISCUTERE, INCUTERE
N;I(');N;S(t'ev,ss) (2):	DISCUTERE, INCUTERE
N;D(idʒ);N;S(ev,att) (2):	ESIGERE, REDIGERE
N;D(idʒ);N;S(ev,ɛtt) (2):	DIRIGERE, PREDILIGERE
N;D(olv);N;S(ev,ɔlt) (2):	ASSOLVERE, RISOLVERE
N;D(ond);N;S(ev,ɔst) (2):	NASCONDERE, RISPONDERE
N;D(oʎʎ);N;S(ev,ɔlt) (2):	COGLIERE, TOGLIERE
N;D(utʃ);N;S(ev,ott) (2):	ADDURRE, CONDURRE
N;D(iddʒ);N;S(ev,itt) (2):	AFFLIGGERE, FRIGGERE
N;D(iŋgw);N;S(ev,int) (2):	DISTINGUERE, ESTINGUERE
N;I(');N;S(ŋgw'ev,nt) (2):	DISTINGUERE, ESTINGUERE
N;S(o,'ɔ);N;S(v'ev,t) (2):	ASSOLVERE, RISOLVERE
N;S(u,'o);N;S(ʃ'ev,t) (2):	ADDURRE, CONDURRE
N;S(o,'ɔ);N;S(dʒ'ev,t) (2):	PORGERE, VOLGERE
N;S(ev,s) (1):	VALERE
N;S(ev,it) (1):	ESISTERE
N;S(ev,tt) (1):	TRARRE
N;S(iv,ut) (1):	VENIRE
N;S(r'iv,'ɛrt) (1):	APRIRE
N;S(an'ev,'ast) (1):	RIMANERE
N;S(ed'ev,'ist) (1):	VEDERE
N;S(ed'ev,'ɛss) (1):	CONCEDERE

N;S(ed'ev,'ɛst) (1):	CHIEDERE
N;S(im'ev,'ɛss) (1):	ESPRIMERE
N;S(iv'ev,'itt) (1):	SCRIVERE
N;S(on'ev,'ɔst) (1):	PORRE
N;S(or'iv,'ɔrt) (1):	MORIRE
N;S(ov'ev,'ɔss) (1):	MUOVERE
N;S(um'ev,'unt) (1):	ASSUMERE
N;D(ed);N;S(v,z) (1):	LEDERE
N;I(');N;D(ʃ'ev) (1):	VINCERE
N;S(eɲɲ'ev,'ɛnt) (1):	SPEGNERE
N;S(ol'ev,'ɔlit) (1):	SOLERE
N;S(omp'ev,'ott) (1):	ROMPERE
N;S(ord'ev,'ɔrs) (1):	MORDERE
N;S(wov'ev,'ɔss) (1):	MUOVERE
N;D(v);N;S(ev,ut) (1):	DEVOLVERE
N;I(');N;S('ev,s) (1):	PARERE
N;I(');N;S('iv,s) (1):	APPARIRE
N;I(j);N;S(iv,ut) (1):	COMPIERE
N;S(eddʒ'ev,'ɛtt) (1):	LEGGERE
N;S(iddʒ'ev,'iss) (1):	AFFIGGERE
N;S(oldʒ'ev,'ɔlt) (1):	VOLGERE
N;S(ordʒ'ev,'ɔrt) (1):	PORGERE
N;S(uddʒ'ev,'utt) (1):	DISTRUGGERE
N;D(eʎʎ);N;S(v,lt) (1):	SCEGLIERE
N;D(itʃ);N;S(v,tt) (1):	DIRE
N;D(od);N;S(ev,ɔz) (1):	RODERE
N;D(r);N;S(iv,ɛrt) (1):	APRIRE
N;I(');N;S(d'ev,s) (1):	ARDERE
N;I(');N;S(r'ev,s) (1):	CORRERE
N;I(st);N;S(ɛr,at) (1):	ESSERE
N;D(an);N;S(ev,ast) (1):	RIMANERE
N;D(aʃʃ);N;S(ev,at) (1):	NASCERE
N;D(ed);N;S(ev,ist) (1):	VEDERE
N;D(ed);N;S(ev,ɛss) (1):	CONCEDERE
N;D(ed);N;S(ev,ɛst) (1):	CHIEDERE
N;D(im);N;S(ev,ɛss) (1):	ESPRIMERE
N;D(indʒ);N;S(v,tt) (1):	STRINGERE
N;D(iv);N;S(ev,itt) (1):	SCRIVERE
N;D(on);N;S(ev,ɔst) (1):	PORRE
N;D(ond);N;S(ev,uz) (1):	FONDERE
N;D(or);N;S(iv,ɔrt) (1):	MORIRE
N;D(ov);N;S(ev,ɔss) (1):	MUOVERE
N;D(um);N;S(ev,unt) (1):	ASSUMERE
N;I(');N;S(dʒ'ev,s) (1):	SPARGERE
N;I(');N;S(m'ev,nt) (1):	ASSUMERE
N;I(');N;S(n'ev,st) (1):	RIMANERE
N;I(');N;S(v'ev,tt) (1):	SCRIVERE
N;S(o,'ɔ);N;D(ʃ'ev) (1):	TORCERE
N;D(eɲɲ);N;S(ev,ɛnt) (1):	SPEGNERE
N;D(omp);N;S(ev,ott) (1):	ROMPERE
N;D(ord);N;S(ev,ɔrs) (1):	MORDERE
N;D(wov);N;S(ev,ɔss) (1):	MUOVERE
N;I(');N;S(mp'ev,tt) (1):	ROMPERE
N;S(o,'ɔ);N;S('iv,t) (1):	MORIRE
N;S(v,ss);N;S(ev,ut) (1):	VIVERE
N;S(w,');N;S(ʃ'ev,t) (1):	CUOCERE
N;D(eddʒ);N;S(ev,ɛtt) (1):	LEGGERE

N;D(iddʒ);N;S(ev,iss) (1):	AFFIGGERE
N;D(oldʒ);N;S(ev,ɔlt) (1):	VOLGERE
N;D(ordʒ);N;S(ev,ɔrt) (1):	PORGERE
N;D(uddʒ);N;S(ev,utt) (1):	DISTRUGGERE
N;I(');N;S(ddʒ'ev,ss) (1):	AFFIGGERE
N;S(el,'u);N;S('ev,s) (1):	ESPELLERE
N;S(o,'ɔ);N;S('ev,it) (1):	SOLERE
N;S(o,'ɔ);N;S(d'ev,s) (1):	MORDERE
N;S(or,'o);N;S('ev,s) (1):	CORRERE
N;D(di);N;S(ʒ'ev,'att) (1):	REDIGERE
N;I(');N;D(r);N;S('ev,s) (1):	CORRERE
N;D(di);N;D(ʒ);N;S(ev,att) (1):	REDIGERE
N;S(e,i);N;D(indʒ);N;S(v,tt) (1):	RESTRINGERE

V.MC_10 (V.IND.PREM.2S) → V.MC_21 (V.PART.PASS.M.S)

N;S(e,ut) (26):	AVERE, BATTERE, BERE, CADERE, CEDERE, CONOSCERE, CREDERE, CRESCERE, DOLERE, DOVERE, GEMERE, GODERE, PASCERE, PERDERE, PIACERE, POSSEDERE, PROVVEDERE, RICEVERE, RIFLETTERE, SAPERE, SEDERE, TACERE, TEMERE, TENERE, VEDERE, VOLERE
N;I(t) (25):	ABBANDONARE, ABOLIRE, AGGREDIRE, AMARE, ANDARE, CAMBIARE, CAPIRE, CARPIRE, CONSUMARE, CONTROLLARE, CONVERTIRE, COPIARE, COSTRUIRE, CREPARE, DOMARE, FERIRE, FREGARE, FUGGIRE, MASTICARE, MERITARE, OSARE, SALIRE, SENTIRE, UDIRE, USCIRE
N;D(end);N;I(z) (5):	ACCENDERE, DIFENDERE, PRENDERE, SCENDERE, TENDERE
N;I(');N;S(dʒ'e,t) (4):	CINGERE, DIPINGERE, INFRANGERE, PUNGERE
N;S(ett'e,'ess) (3):	CONNETTERE, EMETTERE, METTERE
N;S(wot'e,'ɔss) (3):	PERCUOTERE, RISCUOTERE, SCUOTERE
N;D(ett);N;I(ss) (3):	CONNETTERE, EMETTERE, METTERE
N;I(');N;S(tt'e,ss) (3):	CONNETTERE, EMETTERE, METTERE
N;I(');N;S(ddʒ'e,tt) (3):	AFFLIGGERE, DISTRUGGERE, FRIGGERE
N;S(e,at) (2):	DARE, STARE
N;S(ud'e,'uz) (2):	ALLUDERE, CHIUDERE
N;S(ut'e,'uss) (2):	DISCUTERE, INCUTERE
N;S(idʒ'e,'att) (2):	ESIGERE, REDIGERE
N;S(idʒ'e,'ɛtt) (2):	DIRIGERE, PREDILIGERE
N;S(olv'e,'ɔlt) (2):	ASSOLVERE, RISOLVERE
N;S(ond'e,'ɔst) (2):	NASCONDERE, RISPONDERE
N;S(oʎʎ'e,'ɔlt) (2):	COGLIERE, TOGLIERE
N;S(utʃ'e,'ott) (2):	ADDURRE, CONDURRE
N;S(iddʒ'e,'itt) (2):	AFFLIGGERE, FRIGGERE
N;S(iŋgw'e,'int) (2):	DISTINGUERE, ESTINGUERE
N;D(ud);N;S(e,uz) (2):	ALLUDERE, CHIUDERE
N;I(');N;S(d'e,z) (2):	ALLUDERE, CHIUDERE
N;I(');N;S(ʃ'e,t) (2):	CUOCERE, FARE
N;I(');N;S(t'e,ss) (2):	DISCUTERE, INCUTERE
N;I(');N;S(ŋgw'e,nt) (2):	DISTINGUERE, ESTINGUERE
N;S(o,'ɔ);N;S(v'e,t) (2):	ASSOLVERE, RISOLVERE
N;S(u,'o);N;S(ʃ'e,t) (2):	ADDURRE, CONDURRE
N;S(o,'ɔ);N;S(dʒ'e,t) (2):	PORGERE, VOLGERE

N;S(e,s) (1):	VALERE
N;S(e,it) (1):	ESISTERE
N;S(e,tt) (1):	TRARRE
N;S(i,ut) (1):	VENIRE
N;S(v'e,'ut) (1):	DEVOLVERE
N;S(od'e,'ɔz) (1):	RODERE
N;S(r'i,'ɛrt) (1):	APRIRE
N;D(ed);N;I(z) (1):	LEDERE
N;S(an'e,'ast) (1):	RIMANERE
N;S(aʃʃ'e,'at) (1):	NASCERE
N;S(ed'e,'ist) (1):	VEDERE
N;S(ed'e,'ɛss) (1):	CONCEDERE
N;S(ed'e,'ɛst) (1):	CHIEDERE
N;S(im'e,'ɛss) (1):	ESPRIMERE
N;S(iv'e,'itt) (1):	SCRIVERE
N;S(on'e,'ɔst) (1):	PORRE
N;S(ond'e,'uz) (1):	FONDERE
N;S(or'i,'ɔrt) (1):	MORIRE
N;S(ov'e,'ɔss) (1):	MUOVERE
N;S(um'e,'unt) (1):	ASSUMERE
N;I(');N;D(ʃ'e) (1):	VINCERE
N;S(eɲɲ'e,'ɛnt) (1):	SPEGNERE
N;S(eʎʎ'e,'elt) (1):	SCEGLIERE
N;S(itʃ'e,'ett) (1):	DIRE
N;S(omp'e,'ott) (1):	ROMPERE
N;S(ord'e,'ɔrs) (1):	MORDERE
N;S(wov'e,'ɔss) (1):	MUOVERE
N;D(eʎʎ);N;I(lt) (1):	SCEGLIERE
N;D(itʃ);N;I(tt) (1):	DIRE
N;D(v);N;S(e,ut) (1):	DEVOLVERE
N;I(');N;S('e,s) (1):	PARERE
N;I(');N;S('i,s) (1):	APPARIRE
N;I(j);N;S(i,ut) (1):	COMPIERE
N;S(eddʒ'e,'ɛtt) (1):	LEGGERE
N;S(iddʒ'e,'iss) (1):	AFFIGGERE
N;S(indʒ'e,'ett) (1):	STRINGERE
N;S(oldʒ'e,'ɔlt) (1):	VOLGERE
N;S(ordʒ'e,'ɔrt) (1):	PORGERE
N;S(uddʒ'e,'utt) (1):	DISTRUGGERE
N;D(indʒ);N;I(tt) (1):	STRINGERE
N;D(od);N;S(e,ɔz) (1):	RODERE
N;D(r);N;S(i,ɛrt) (1):	APRIRE
N;I(');N;S(d'e,s) (1):	ARDERE
N;I(');N;S(r'e,s) (1):	CORRERE
N;I('ɛ);N;S('i,t) (1):	APRIRE
N;D(aʃʃ);N;S(e,at) (1):	NASCERE
N;D(ond);N;S(e,uz) (1):	FONDERE
N;I(');N;S(dʒ'e,s) (1):	SPARGERE
N;I(');N;S(m'e,nt) (1):	ASSUMERE
N;I(');N;S(n'e,st) (1):	RIMANERE
N;I(');N;S(v'e,tt) (1):	SCRIVERE
N;I(');N;S(ʃʃ'e,t) (1):	NASCERE
N;S(o,'ɔ);N;D(ʃ'e) (1):	TORCERE
N;I(');N;S(mp'e,tt) (1):	ROMPERE
N;I(');N;S(ʎʎ'e,lt) (1):	SCEGLIERE
N;S(f,st);N;S(o,at) (1):	ESSERE
N;S(o,'ɔ);N;S('i,t) (1):	MORIRE

N;S(v,ss);N;S(e,ut) (1):	VIVERE
N;S(w,');N;S(ʃ'e,t) (1):	CUOCERE
N;I(');N;S(ddʒ'e,ss) (1):	AFFIGGERE
N;S(el,'u);N;S('e,s) (1):	ESPELLERE
N;S(i,'e);N;S(ʃ'e,t) (1):	DIRE
N;S(o,'ɔ);N;S('e,it) (1):	SOLERE
N;S(o,'ɔ);N;S(d'e,s) (1):	MORDERE
N;S(or,'o);N;S('e,s) (1):	CORRERE
N;I(');N;D(r);N;S('e,s) (1):	CORRERE
N;S(e,i);N;S(indʒ'e,'ett) (1):	RESTRINGERE
N;S(e,i);N;D(indʒ);N;I(tt) (1):	RESTRINGERE

V.IND.PREM.3S → V.IND.PREM.1S

N;S(e,i) (99):	ACCENDERE, ADDURRE, AFFIGGERE, AFFLIGGERE, ALLUDERE, APPARIRE, ARDERE, ASSOLVERE, ASSUMERE, AVERE, BERE, CADERE, CEDERE, CHIEDERE, CHIUDERE, CINGERE, COGLIERE, CONCEDERE, CONDURRE, CONOSCERE, CORRERE, CREDERE, CRESCERE, CUOCERE, DARE, DEVOLVERE, DIFENDERE, DIPINGERE, DIRE, DIRIGERE, DISCUTERE, DISTINGUERE, DISTRUGGERE, DOLERE, DOVERE, EMETTERE, ESPELLERE, ESPRIMERE, ESTINGUERE, FARE, FONDERE, FRIGGERE, GEMERE, INCUTERE, INFRANGERE, LEDERE, LEGGERE, METTERE, MORDERE, MUOVERE, NASCERE, NASCONDERE, PARERE, PASCERE, PERCUOTERE, PERDERE, PIACERE, PORGERE, PORRE, POSSEDERE, PREDILIGERE, PRENDERE, PROVVEDERE, PUNGERE, REDIGERE, RESTRINGERE, RICEVERE, RIMANERE, RISCUOTERE, RISOLVERE, RISPONDERE, RODERE, ROMPERE, SAPERE, SCEGLIERE, SCENDERE, SCRIVERE, SCUOTERE, SEDERE, SPARGERE, SPEGNERE, SPLENDERE, STARE, STRIDERE, STRINGERE, TACERE, TEMERE, TENDERE, TENERE, TOGLIERE, TORCERE, TRARRE, VALERE, VEDERE, VENIRE, VINCERE, VIVERE, VOLERE, VOLGERE
N;I(j) (36):	ABOLIRE, AGGREDIRE, APRIRE, BATTERE, CAPIRE, CARPIRE, CEDERE, COMPIERE, CONCERNERE, CONNETTERE, CONVERTIRE, COSTRUIRE, CREDERE, DEVOLVERE, DOVERE, ESIGERE, ESISTERE, ESSERE, FERIRE, FUGGIRE, GEMERE, GODERE, MORIRE, PASCERE, POSSEDERE, RICEVERE, RIFLETTERE, SALIRE, SEDERE, SENTIRE, SOLERE, SPLENDERE, STRIDERE, TEMERE, UDIRE, USCIRE
N;S(ɔ,aj) (13):	ABBANDONARE, AMARE, ANDARE, CAMBIARE, CONSUMARE, CONTROLLARE, COPIARE, CREPARE, DOMARE, FREGARE, MASTICARE, MERITARE, OSARE
N;I(tti) (12):	CEDERE, CREDERE, DEVOLVERE, DOVERE, GEMERE, PASCERE, POSSEDERE, RICEVERE, SEDERE, SPLENDERE, STRIDERE, TEMERE

N;S(tte,j) (12):	CEDERE, CREDERE, DEVOLVERE, DOVERE, GEMERE, PASCERE, POSSEDERE, RICEVERE, SEDERE, SPLENDERE, STRIDERE, TEMERE

V.MC_4 (V.IND.PRES.1P) → V.MC_7 (V.IND.IMPF.1S)

N;S(j,'ev) (62):	ACCENDERE, ALLUDERE, ARDERE, ASSOLVERE, ASSUMERE, BATTERE, BERE, CADERE, CEDERE, CHIEDERE, CHIUDERE, CONCEDERE, CONCERNERE, CONNETTERE, CORRERE, CREDERE, DEVOLVERE, DIFENDERE, DISCUTERE, DISTINGUERE, DOLERE, EMETTERE, ESISTERE, ESPELLERE, ESPRIMERE, ESTINGUERE, FONDERE, GEMERE, GODERE, INCUTERE, LEDERE, METTERE, MORDERE, MUOVERE, NASCONDERE, PERCUOTERE, PERDERE, PORRE, POSSEDERE, PRENDERE, PROVVEDERE, RICEVERE, RIFLETTERE, RIMANERE, RISCUOTERE, RISOLVERE, RISPONDERE, RODERE, ROMPERE, SCENDERE, SCRIVERE, SCUOTERE, SEDERE, SPLENDERE, STRIDERE, TEMERE, TENDERE, TENERE, TRARRE, VALERE, VEDERE, VIVERE
N;I('ev) (34):	ADDURRE, AFFIGGERE, AFFLIGGERE, CINGERE, COGLIERE, CONDURRE, CONOSCERE, CRESCERE, CUOCERE, DIPINGERE, DIRE, DIRIGERE, DISTRUGGERE, ESIGERE, FRIGGERE, INFRANGERE, LEGGERE, NASCERE, PASCERE, PIACERE, PORGERE, PREDILIGERE, PUNGERE, REDIGERE, RESTRINGERE, SCEGLIERE, SPARGERE, SPEGNERE, STRINGERE, TACERE, TOGLIERE, TORCERE, VINCERE, VOLGERE
N;S(j,'iv) (15):	ABOLIRE, AGGREDIRE, APPARIRE, APRIRE, CAPIRE, CARPIRE, COMPIERE, CONVERTIRE, COSTRUIRE, FERIRE, MORIRE, SALIRE, SENTIRE, UDIRE, VENIRE
N;S(j,'av) (13):	ABBANDONARE, AMARE, ANDARE, CONSUMARE, CONTROLLARE, CREPARE, DARE, DOMARE, FREGARE, MASTICARE, MERITARE, OSARE, STARE
N;I('av) (2):	CAMBIARE, COPIARE
N;I('iv) (2):	FUGGIRE, USCIRE
N;S(ʎʎ,l'ev) (2):	SOLERE, VOLERE
N;S(bbj,v'ev) (2):	AVERE, DOVERE
N;S(j,r'ev) (1):	PARERE
N;S(pj,'ev) (1):	SAPERE
N;S(sj,'ɛr) (1):	ESSERE
N;S(tʃ,ʃ'ev) (1):	FARE
N;D(t);N;I('ev) (1):	FARE
N;D(w);N;I('ev) (1):	CUOCERE
N;I(w);N;I('ev) (1):	CUOCERE
N;D(w);N;S(j,'ev) (1):	MUOVERE
N;I(w);N;S(j,'ev) (1):	MUOVERE

V.MC_4 (V.IND.PRES.1P) → V.MC_14 (V.IND.FUT.1S)

N;S(j,e) (61):	ABBANDONARE, ACCENDERE, ALLUDERE, AMARE, ARDERE, ASSOLVERE, ASSUMERE, BATTERE, CEDERE, CHIEDERE, CHIUDERE, CONCEDERE, CONCERNERE, CONNETTERE, CONSUMARE, CONTROLLARE, CORRERE, CREDERE, CREPARE, DEVOLVERE, DIFENDERE, DISCUTERE, DISTINGUERE, DOMARE, EMETTERE, ESISTERE, ESPELLERE, ESPRIMERE, ESTINGUERE, FONDERE, FREGARE, GEMERE, INCUTERE, LEDERE, MASTICARE, MERITARE, METTERE, MORDERE, MUOVERE, NASCONDERE, OSARE, PERCUOTERE, PERDERE, POSSEDERE, PRENDERE, PROVVEDERE, RICEVERE, RIFLETTERE, RISCUOTERE, RISOLVERE, RISPONDERE, RODERE, ROMPERE, SCENDERE, SCRIVERE, SCUOTERE, SEDERE, SPLENDERE, STRIDERE, TEMERE, TENDERE
N;I(e) (33):	AFFIGGERE, AFFLIGGERE, CAMBIARE, CINGERE, COGLIERE, CONOSCERE, COPIARE, CRESCERE, CUOCERE, DIPINGERE, DIRIGERE, DISTRUGGERE, ESIGERE, FRIGGERE, INFRANGERE, LEGGERE, NASCERE, PASCERE, PIACERE, PORGERE, PREDILIGERE, PUNGERE, REDIGERE, RESTRINGERE, SCEGLIERE, SPARGERE, SPEGNERE, STRINGERE, TACERE, TOGLIERE, TORCERE, VINCERE, VOLGERE
N;S(j,i) (14):	ABOLIRE, AGGREDIRE, APPARIRE, APRIRE, CAPIRE, CARPIRE, COMPIERE, CONVERTIRE, COSTRUIRE, FERIRE, MORIRE, SALIRE, SENTIRE, UDIRE
N;D(j) (5):	ANDARE, CADERE, GODERE, VEDERE, VIVERE
N;S(nj,r) (4):	PORRE, RIMANERE, TENERE, VENIRE
N;S(j,a) (3):	DARE, ESSERE, STARE
N;I(i) (2):	FUGGIRE, USCIRE
N;S(j,r) (2):	PARERE, TRARRE
N;S(lj,r) (2):	DOLERE, VALERE
N;S(tʃ,r) (2):	ADDURRE, CONDURRE
N;S(bbj,v) (2):	AVERE, DOVERE
N;D(pj) (1):	SAPERE
N;D(tʃ) (1):	DIRE
N;D(ttʃ) (1):	FARE
N;S(vj,r) (1):	BERE
N;S(ʎʎ,r) (1):	VOLERE
N;D(p);N;D(j) (1):	SAPERE
N;D(w);N;I(e) (1):	CUOCERE
N;I(w);N;I(e) (1):	CUOCERE
N;D(w);N;S(j,e) (1):	MUOVERE
N;I(w);N;S(j,e) (1):	MUOVERE

V.MC_7 (V.IND.IMPF.1S) → V.GERU

N;D(v) (118):	ABBANDONARE, ACCENDERE, ADDURRE, AFFIGGERE, AFFLIGGERE, ALLUDERE, AMARE, ANDARE, ARDERE, ASSOLVERE, ASSUMERE,

	AVERE, BATTERE, BERE, CADERE, CAMBIARE, CEDERE, CHIEDERE, CHIUDERE, CINGERE, COGLIERE, CONCEDERE, CONCERNERE, CONDURRE, CONNETTERE, CONOSCERE, CONSUMARE, CONTROLLARE, COPIARE, CORRERE, CREDERE, CREPARE, CRESCERE, CUOCERE, DARE, DEVOLVERE, DIFENDERE, DIPINGERE, DIRE, DIRIGERE, DISCUTERE, DISTINGUERE, DISTRUGGERE, DOLERE, DOMARE, DOVERE, EMETTERE, ESIGERE, ESISTERE, ESPELLERE, ESPRIMERE, ESTINGUERE, FARE, FONDERE, FREGARE, FRIGGERE, GEMERE, GODERE, INCUTERE, INFRANGERE, LEDERE, LEGGERE, MASTICARE, MERITARE, METTERE, MORDERE, MUOVERE, NASCERE, NASCONDERE, OSARE, PARERE, PASCERE, PERCUOTERE, PERDERE, PIACERE, PORGERE, PORRE, POSSEDERE, PREDILIGERE, PRENDERE, PROVVEDERE, PUNGERE, REDIGERE, RESTRINGERE, RICEVERE, RIFLETTERE, RIMANERE, RISCUOTERE, RISOLVERE, RISPONDERE, RODERE, ROMPERE, SAPERE, SCEGLIERE, SCENDERE, SCRIVERE, SCUOTERE, SEDERE, SOLERE, SPARGERE, SPEGNERE, SPLENDERE, STARE, STRIDERE, STRINGERE, TACERE, TEMERE, TENDERE, TENERE, TOGLIERE, TORCERE, TRARRE, VALERE, VEDERE, VINCERE, VIVERE, VOLERE, VOLGERE
N;S(iv,e) (16):	ABOLIRE, AGGREDIRE, APPARIRE, APRIRE, CAPIRE, CARPIRE, CONVERTIRE, COSTRUIRE, FERIRE, FUGGIRE, MORIRE, SALIRE, SENTIRE, UDIRE, USCIRE, VENIRE
N;D(w);N;D(v) (2):	CUOCERE, MUOVERE
N;I(w);N;D(v) (2):	CUOCERE, MUOVERE
N;S('iv,j'e) (1):	COMPIERE
N;I(ess);N;D(r) (1):	ESSERE
N;I(j);N;S(iv,e) (1):	COMPIERE

V.MC_7 (V.IND.IMPF.1S) → V.INF

N;I(');N;D(');N;D(v) (56):	ACCENDERE, AFFIGGERE, AFFLIGGERE, ALLUDERE, ARDERE, ASSUMERE, BATTERE, CEDERE, CHIEDERE, CHIUDERE, CINGERE, CONCEDERE, CONNETTERE, CONOSCERE, CORRERE, CREDERE, CRESCERE, DIFENDERE, DIPINGERE, DIRIGERE, DISCUTERE, DISTINGUERE, DISTRUGGERE, EMETTERE, ESIGERE, ESISTERE, ESPRIMERE, ESTINGUERE, FONDERE, FRIGGERE, GEMERE, INCUTERE, INFRANGERE, LEDERE, METTERE, NASCERE, NASCONDERE, PASCERE, PREDILIGERE, PUNGERE, REDIGERE, RESTRINGERE, RICEVERE, RISPONDERE, ROMPERE, SCEGLIERE, SCENDERE, SCRIVERE, SPARGERE,

	SPEGNERE, SPLENDERE, STRIDERE, STRINGERE, TENDERE, VINCERE, VIVERE
N;D(v) (51):	ABBANDONARE, ABOLIRE, AGGREDIRE, AMARE, ANDARE, APPARIRE, APRIRE, AVERE, CADERE, CAMBIARE, CAPIRE, CARPIRE, CONSUMARE, CONTROLLARE, CONVERTIRE, COPIARE, COSTRUIRE, CREPARE, DARE, DOLERE, DOMARE, DOVERE, FERIRE, FREGARE, FUGGIRE, GODERE, MASTICARE, MERITARE, MORIRE, OSARE, PARERE, PIACERE, POSSEDERE, PROVVEDERE, RIFLETTERE, RIMANERE, SALIRE, SAPERE, SEDERE, SENTIRE, SOLERE, STARE, TACERE, TEMERE, TENERE, UDIRE, USCIRE, VALERE, VEDERE, VENIRE, VOLERE
N;S(o,'ɔ);N;D(');N;D(v) (15):	ASSOLVERE, COGLIERE, CUOCERE, DEVOLVERE, MORDERE, MUOVERE, PERCUOTERE, PORGERE, RISCUOTERE, RISOLVERE, RODERE, SCUOTERE, TOGLIERE, TORCERE, VOLGERE
N;S(e,'ɛ);N;D(');N;D(v) (5):	CONCERNERE, ESPELLERE, LEGGERE, PERDERE, PRENDERE
N;S(ot','ɔt);N;D(v) (3):	PERCUOTERE, RISCUOTERE, SCUOTERE
N;D(utʃ);N;S(ev,ur) (2):	ADDURRE, CONDURRE
N;S(oʌ,'ɔ);N;S(',ʌ);N;D(v) (2):	COGLIERE, TOGLIERE
N;S(r,sse) (1):	ESSERE
N;D(ev);N;D(v) (1):	BERE
N;D(ev);N;I(e) (1):	RICEVERE
N;D(a);N;S(ev,ar) (1):	TRARRE
N;D(atʃ);N;S(ev,a) (1):	FARE
N;D(itʃ);N;S(ev,i) (1):	DIRE
N;D(on);N;S(ev,or) (1):	PORRE
N;I(');N;S('iv,je) (1):	COMPIERE
N;S(od','ɔd);N;D(v) (1):	RODERE
N;S(ov','ɔv);N;D(v) (1):	MUOVERE
N;D(ov);N;S(e,ɔ);N;I(e) (1):	MUOVERE
N;S(o,w'ɔ);N;D(');N;D(v) (1):	CUOCERE
N;S(ov,w);N;S(e,ɔ);N;I(e) (1):	MUOVERE
N;S(el,'ɛ);N;S(',l);N;D(v) (1):	ESPELLERE

Table 15 below shows all the relations computed. For each pair <A, B> of metacells only the relation A→B is given; the relation B→A is symmetrical (replacing Insertion with Deletion and *vice versa*, and exchanging the terms of Substitution). The table is sorted, in decreasing order, by the number of lexemes that implement each relation.

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.MC_10	V.IND.PREM.1P	N	135	AMARE
V.MC_7	V.MC_8	N;D(');N;D(v)	135	AMARE
V.GERU	V.MC_20	N	134	AMARE
V.IND.PRES.2P	V.MC_8	N;D(');N	133	AMARE
V.IND.PRES.2P	V.MC_7	N;I(v)	133	AMARE
V.IND.PRES.2P	V.IMP.2P	N	131	AMARE
V.MC_7	V.IMP.2P	N;D(v)	129	AMARE
V.MC_8	V.IMP.2P	N;I(');N	129	AMARE
V.IND.PRES.1S	V.MC_15	N;S(o,a)	119	ANDARE
V.IND.PRES.2S	V.IMP.2S	N	118	ANDARE
V.IND.PRES.1S	V.IND.PRES.3P	N	117	ARDERE
V.IND.PRES.3P	V.MC_15	N;S(o,a)	117	ARDERE
V.IND.PRES.2S	V.IND.PRES.3S	N;S(i,e)	115	ARDERE
V.IND.PRES.3S	V.IMP.2S	N;S(e,i)	113	ARDERE
V.MC_8	V.INF	N;I(');N	106	AMARE
V.IND.PRES.2P	V.MC_14	N;D(');N	102	ARDERE
V.MC_14	V.IMP.2P	N;I(');N	102	ARDERE
V.MC_7	V.MC_14	N;D(');N;D(v)	100	ARDERE
V.MC_8	V.MC_14	N	100	ARDERE
V.IND.PREM.1S	V.IND.PREM.3S	N;S(i,e)	99	ARDERE
V.IND.PREM.1S	V.IND.PREM.3P	N;S(i,er)	99	ARDERE
V.IND.PREM.3S	V.IND.PREM.3P	N;I(r)	99	ARDERE
V.MC_14	V.INF	N;I(');N	84	STARE
V.MC_4	V.GERU	N;S(j,'e)	76	ARDERE
V.MC_4	V.MC_20	N;S(j,'e)	76	ARDERE
V.IND.PRES.2P	V.INF	N;I(');N;D(');N	55	CINGERE
V.MC_7	V.INF	N;I(');N;D(');N;D(v)	55	CINGERE
V.IMP.2P	V.INF	N;I(');N;D(');N	55	CINGERE
V.IND.PRES.2P	V.INF	N	53	STARE
V.MC_7	V.INF	N;D(v)	51	STARE
V.IND.PREM.1S	V.IND.PREM.3P	N;S(j,ron)	49	AMARE
V.IMP.2P	V.INF	N	49	STARE
V.MC_4	V.GERU	N;I('e)	37	PIACERE
V.MC_4	V.MC_20	N;I('e)	37	PIACERE
V.IND.PREM.1S	V.IND.PREM.3S	N;D(j)	36	APRIRE
V.IND.PREM.3S	V.IND.PREM.3P	N;I(ron)	36	APRIRE
V.IND.PREM.1S	V.MC_21	N;S(j,t)	25	AMARE
V.IND.PREM.3P	V.MC_21	N;S(ron,t)	25	AMARE
V.IND.PREM.1S	V.MC_21	N;D(i)	23	ESPRIMERE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.IND.PREM.3S	V.MC_21	N;D(e)	23	ESPRIMERE
V.IND.PREM.3P	V.MC_21	N;D(er)	23	ESPRIMERE
V.IND.PREM.1S	V.MC_21	N;S(si,t)	17	INFRANGERE
V.IND.PREM.3S	V.MC_21	N;S(se,t)	17	INFRANGERE
V.IND.PREM.3P	V.MC_21	N;S(ser,t)	17	INFRANGERE
V.IND.PRES.3S	V.IMP.2S	N	16	STARE
V.IND.PRES.2P	V.INF	N;S(o,'ɔ);N;D(');N	15	TORCERE
V.MC_7	V.INF	N;S(o,'ɔ);N;D(');N;D(v)	15	TORCERE
V.MC_8	V.INF	N;S(o,'ɔ);N	15	TORCERE
V.MC_14	V.INF	N;S(o,'ɔ);N	15	TORCERE
V.IMP.2P	V.INF	N;S(o,'ɔ);N;D(');N	15	TORCERE
V.MC_4	V.GERU	N;S(j,'a)	13	STARE
V.MC_4	V.MC_20	N;S(j,'a)	13	STARE
V.IND.PREM.1S	V.IND.PREM.3S	N;S(aj,ɔ)	13	AMARE
V.IND.PREM.3S	V.IND.PREM.3P	N;S(ɔ,aron)	13	AMARE
V.IND.PREM.3S	V.MC_21	N;S(ɔ,at)	13	AMARE
V.IND.PRES.1S	V.IND.PRES.3P	N;S(o,a)	12	CONSUMARE
V.IND.PRES.2P	V.MC_14	N;S('a,e)	12	CONSUMARE
V.MC_7	V.MC_14	N;S('av,e)	12	CONSUMARE
V.MC_8	V.MC_14	N;S(a,e)	12	CONSUMARE
V.IND.PREM.1S	V.IND.PREM.3S	N;D(tti)	12	CEDERE
V.IND.PREM.1S	V.IND.PREM.3S	N;S(j,tte)	12	CEDERE
V.IND.PREM.1S	V.IND.PREM.3P	N;S(j,tter)	12	CEDERE
V.IND.PREM.1S	V.IND.PREM.3P	N;S(tti,ron)	12	CEDERE
V.IND.PREM.1S	V.IND.PREM.3P	N;I(ett);N;S(j,r)	12	CEDERE
V.IND.PREM.1S	V.MC_21	N;S(ej,ut)	12	CEDERE
V.IND.PREM.3S	V.IND.PREM.3P	N;I(tter)	12	CEDERE
V.IND.PREM.3S	V.IND.PREM.3P	N;S(tte,ron)	12	CEDERE
V.IND.PREM.3S	V.IND.PREM.3P	N;I(ett);N;I(r)	12	CEDERE
V.IND.PREM.3S	V.MC_21	N;I(t)	12	ABOLIRE
V.IND.PREM.3S	V.MC_21	N;S(e,ut)	12	CEDERE
V.IND.PREM.3P	V.MC_21	N;S(eron,ut)	12	CEDERE
V.MC_14	V.IMP.2P	N;S(e,'a)	12	CONSUMARE
V.MC_14	V.INF	N;S(e,'a)	12	CONSUMARE
V.IND.PRES.1S	V.MC_15	N;S(o,i)	10	CONSUMARE
V.IND.PRES.2S	V.IND.PRES.3S	N;S(i,a)	10	CONSUMARE
V.IND.PRES.2S	V.IMP.2S	N;S(i,a)	10	CONSUMARE
V.IND.PRES.3P	V.MC_15	N;S(a,i)	10	CONSUMARE
V.IND.PREM.1S	V.MC_21	N;S(e,u);N;D(ti)	9	CEDERE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.IND.PREM.1S	V.MC_21	N;S(et,u);N;D(i)	9	CEDERE
V.IND.PREM.3S	V.MC_21	N;S(e,u);N;D(te)	9	CEDERE
V.IND.PREM.3S	V.MC_21	N;S(et,u);N;D(e)	9	CEDERE
V.IND.PREM.3P	V.MC_21	N;S(e,u);N;D(ter)	9	CEDERE
V.IND.PREM.3P	V.MC_21	N;S(et,u);N;D(er)	9	CEDERE
V.IND.PRES.2P	V.MC_14	N;D('e)	8	VIVERE
V.MC_7	V.MC_14	N;D('ev)	8	VIVERE
V.MC_8	V.MC_14	N;D(e)	8	VIVERE
V.IND.PREM.1S	V.MC_21	N;S(ssi,tt)	8	REDIGERE
V.IND.PREM.3S	V.MC_21	N;S(sse,tt)	8	REDIGERE
V.IND.PREM.3P	V.MC_21	N;S(sser,tt)	8	REDIGERE
V.MC_14	V.INF	N;I('e)	7	PARERE
V.IND.PRES.2S	V.IND.PRES.3S	N;D(j)	6	STARE
V.IND.PRES.2P	V.INF	N;S(e,'ɛ);N;D(');N	5	PERDERE
V.MC_7	V.INF	N;S(e,'ɛ);N;D(');N;D(v)	5	PERDERE
V.MC_8	V.INF	N;S(e,'ɛ);N	5	PERDERE
V.MC_14	V.IMP.2P	N;I('e)	5	VIVERE
V.MC_14	V.INF	N;S(e,'ɛ);N	5	PERDERE
V.IMP.2P	V.INF	N;S(e,'ɛ);N;D(');N	5	PERDERE
V.IND.PRES.1S	V.IND.PRES.3P	N;S(ɔ,an)	4	STARE
V.IND.PRES.2S	V.IMP.2S	N;D(j)	4	STARE
V.IND.PRES.3S	V.IMP.2S	N;I(j)	4	STARE
V.IND.PRES.2P	V.MC_14	N;S(l'e,r)	3	VALERE
V.IND.PRES.2P	V.MC_14	N;S(n'e,r)	3	RIMANERE
V.IND.PRES.2P	V.INF	N;S(ot','ɔt);N	3	PERCUOTERE
V.MC_7	V.MC_14	N;D(v'e);N	3	VIVERE
V.MC_7	V.MC_14	N;S(l'ev,r)	3	VALERE
V.MC_7	V.MC_14	N;S(n'ev,r)	3	RIMANERE
V.MC_7	V.INF	N;S(ot','ɔt);N;D(v)	3	PERCUOTERE
V.MC_8	V.MC_14	N;S(le,r)	3	VALERE
V.MC_8	V.MC_14	N;S(ne,r)	3	RIMANERE
V.IND.PREM.1S	V.MC_21	N;S(ozi,ɔst)	3	NASCONDERE
V.IND.PREM.3S	V.MC_21	N;S(oze,ɔst)	3	NASCONDERE
V.IND.PREM.3P	V.MC_21	N;S(ozer,ɔst)	3	NASCONDERE
V.MC_14	V.IMP.2P	N;S(r,n'e)	3	RIMANERE
V.MC_14	V.INF	N;S(r,l'e)	3	VALERE
V.IMP.2P	V.INF	N;S(ot','ɔt);N	3	PERCUOTERE
V.IND.PRES.1S	V.IND.PRES.3P	N;D(j);N;S(d,gg);N	2	POSSEDERE
V.IND.PRES.1S	V.IND.PRES.3P	N;I(j);N;S(gg,d);N	2	POSSEDERE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.IND.PRES.1S	V.MC_15	N;S(jo,i)	2	COPIARE
V.IND.PRES.1S	V.MC_15	N;S(ɔ,ia)	2	STARE
V.IND.PRES.1S	V.MC_15	N;D(j);N;S(do,gga)	2	POSSEDERE
V.IND.PRES.1S	V.MC_15	N;I(j);N;S(ggo,da)	2	POSSEDERE
V.IND.PRES.2S	V.IND.PRES.3S	N;S(i,ja)	2	COPIARE
V.IND.PRES.2S	V.IMP.2S	N;S(i,ja)	2	COPIARE
V.MC_4	V.GERU	N;I('a)	2	COPIARE
V.MC_4	V.GERU	N;S(ʎʎ,l'e)	2	VOLERE
V.MC_4	V.GERU	N;S(bbj,v'e)	2	AVERE
V.MC_4	V.MC_20	N;I('a)	2	COPIARE
V.MC_4	V.MC_20	N;S(ʎʎ,l'e)	2	VOLERE
V.IND.PRES.2P	V.MC_7	N;D(w);N;I(v)	2	CUOCERE
V.IND.PRES.2P	V.MC_7	N;I(w);N;I(v)	2	CUOCERE
V.IND.PRES.2P	V.MC_8	N;D(');N;I(tʃe)	2	DIRE
V.IND.PRES.2P	V.MC_8	N;D(w);N;D(');N	2	CUOCERE
V.IND.PRES.2P	V.MC_8	N;I(w);N;D(');N	2	CUOCERE
V.IND.PRES.2P	V.MC_14	N;S(tʃ'e,r)	2	CONDURRE
V.IND.PRES.2P	V.MC_14	N;D(w);N;D(');N	2	CUOCERE
V.IND.PRES.2P	V.MC_14	N;I(w);N;D(');N	2	CUOCERE
V.IND.PRES.2P	V.IMP.2P	N;D(w);N	2	CUOCERE
V.IND.PRES.2P	V.IMP.2P	N;I(w);N	2	CUOCERE
V.IND.PRES.2P	V.INF	N;S(utʃ'e,'ur)	2	CONDURRE
V.IND.PRES.2P	V.INF	N;D(utʃ);N;S(e,ur)	2	CONDURRE
V.IND.PRES.2P	V.INF	N;I(');N;S(tʃ'e,r)	2	CONDURRE
V.IND.PRES.2P	V.INF	N;S(o,w'ɔ);N;D(');N	2	CUOCERE
V.IND.PRES.2P	V.INF	N;S(oʎ,'ɔ);N;S(',ʎ);N	2	COGLIERE
V.IND.PRES.3P	V.MC_15	N;S(ja,i)	2	COPIARE
V.IND.PRES.3P	V.MC_15	N;S(an,ia)	2	STARE
V.IND.PRES.3P	V.MC_15	N;I(i);N;D(n)	2	STARE
V.IND.PRES.3P	V.MC_15	N;D(j);N;S(do,gga)	2	POSSEDERE
V.IND.PRES.3P	V.MC_15	N;I(j);N;S(ggo,da)	2	POSSEDERE
V.MC_7	V.MC_8	N;D(w);N;D(');N;D(v)	2	CUOCERE
V.MC_7	V.MC_8	N;I(w);N;D(');N;D(v)	2	CUOCERE
V.MC_7	V.MC_14	N;D(tʃ'ev)	2	DIRE
V.MC_7	V.MC_14	N;S(tʃ'ev,r)	2	CONDURRE
V.MC_7	V.MC_14	N;D(w);N;D(');N;D(v)	2	CUOCERE
V.MC_7	V.MC_14	N;I(w);N;D(');N;D(v)	2	CUOCERE
V.MC_7	V.IMP.2P	N;D(w);N;D(v)	2	CUOCERE
V.MC_7	V.IMP.2P	N;I(w);N;D(v)	2	CUOCERE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.MC_7	V.INF	N;D(utʃ);N;S(ev,ur)	2	CONDURRE
V.MC_7	V.INF	N;S(oʎ,'ɔ);N;S(',ʎ);N;D (v)	2	COGLIERE
V.MC_8	V.MC_14	N;D(tʃe)	2	DIRE
V.MC_8	V.MC_14	N;D(w);N	2	CUOCERE
V.MC_8	V.MC_14	N;I(w);N	2	CUOCERE
V.MC_8	V.MC_14	N;S(tʃe,r)	2	CONDURRE
V.MC_8	V.IMP.2P	N;D(w);N;I(');N	2	CUOCERE
V.MC_8	V.IMP.2P	N;I(');N;D(tʃe)	2	DIRE
V.MC_8	V.IMP.2P	N;I(w);N;I(');N	2	CUOCERE
V.MC_8	V.INF	N;S(o,w'ɔ);N	2	CUOCERE
V.MC_8	V.INF	N;S(utʃe,'ur)	2	CONDURRE
V.MC_8	V.INF	N;I(');N;D(tʃe)	2	DIRE
V.MC_8	V.INF	N;I(');N;S(tʃe,r)	2	CONDURRE
V.IND.PREM.1S	V.MC_21	N;S(vi,s)	2	PARERE
V.IND.PREM.1S	V.MC_21	N;S(izi,ess)	2	METTERE
V.IND.PREM.1S	V.MC_21	N;S(ussi,ott)	2	CONDURRE
V.IND.PREM.1S	V.MC_21	N;S('idi,ed'ut)	2	PROVVEDERE
V.IND.PREM.1S	V.MC_21	N;D(');N;S(ni,'ut)	2	TENERE
V.IND.PREM.1S	V.MC_21	N;S('akkwi,atʃ'ut)	2	TACERE
V.IND.PREM.1S	V.MC_21	N;I(ed);N;S(idi,ut)	2	PROVVEDERE
V.IND.PREM.1S	V.MC_21	N;S('i,e);N;S(i,'ut)	2	PROVVEDERE
V.IND.PREM.1S	V.MC_21	N;D(');N;S(bbi,ʃʃ'ut)	2	CONOSCERE
V.IND.PREM.1S	V.MC_21	N;D(');N;S(kkwi,tʃ'ut)	2	TACERE
V.MC_10	V.IND.PREM.1P	N;D(w);N	2	CUOCERE
V.MC_10	V.IND.PREM.1P	N;I(w);N	2	CUOCERE
V.IND.PREM.3S	V.MC_21	N;S(ve,s)	2	PARERE
V.IND.PREM.3S	V.MC_21	N;S(ize,ess)	2	METTERE
V.IND.PREM.3S	V.MC_21	N;S(usse,ott)	2	CONDURRE
V.IND.PREM.3S	V.MC_21	N;S('ide,ed'ut)	2	PROVVEDERE
V.IND.PREM.3S	V.MC_21	N;D(');N;S(ne,'ut)	2	TENERE
V.IND.PREM.3S	V.MC_21	N;S('akkwe,atʃ'ut)	2	TACERE
V.IND.PREM.3S	V.MC_21	N;I(ed);N;S(ide,ut)	2	PROVVEDERE
V.IND.PREM.3S	V.MC_21	N;S('i,e);N;S(e,'ut)	2	PROVVEDERE
V.IND.PREM.3S	V.MC_21	N;D(');N;S(bbe,ʃʃ'ut)	2	CONOSCERE
V.IND.PREM.3S	V.MC_21	N;D(');N;S(kkwe,tʃ'ut)	2	TACERE
V.IND.PREM.3P	V.MC_21	N;S(ver,s)	2	PARERE
V.IND.PREM.3P	V.MC_21	N;S(izer,ess)	2	METTERE
V.IND.PREM.3P	V.MC_21	N;S(usser,ott)	2	CONDURRE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.IND.PREM.3P	V.MC_21	N;S('ider,ed'ut)	2	PROVVEDERE
V.IND.PREM.3P	V.MC_21	N;D(iz);N;S(r,ss)	2	METTERE
V.IND.PREM.3P	V.MC_21	N;D(');N;S(ner,'ut)	2	TENERE
V.IND.PREM.3P	V.MC_21	N;S('i,e);N;S(er,'ut)	2	PROVVEDERE
V.IND.PREM.3P	V.MC_21	N;D(');N;S(bber,ʃʃ'ut)	2	CONOSCERE
V.IND.PREM.3P	V.MC_21	N;D(');N;S(kkwer,tʃ'ut)	2	TACERE
V.MC_14	V.IMP.2P	N;S(r,l'e)	2	VALERE
V.MC_14	V.IMP.2P	N;S(r,tʃ'e)	2	CONDURRE
V.MC_14	V.IMP.2P	N;D(w);N;I(');N	2	CUOCERE
V.MC_14	V.IMP.2P	N;I(w);N;I(');N	2	CUOCERE
V.MC_14	V.INF	N;S(r,n'e)	2	RIMANERE
V.MC_14	V.INF	N;S(o,w'ɔ);N	2	CUOCERE
V.IMP.2P	V.INF	N;S(utʃ'e,'ur)	2	CONDURRE
V.IMP.2P	V.INF	N;D(utʃ);N;S(e,ur)	2	CONDURRE
V.IMP.2P	V.INF	N;I(');N;S(tʃ'e,r)	2	CONDURRE
V.IMP.2P	V.INF	N;S(o,w'ɔ);N;D(');N	2	CUOCERE
V.IMP.2P	V.INF	N;S(oʎ,'ɔ);N;S(',ʎ);N	2	COGLIERE
V.GERU	V.MC_20	N;D(w);N	2	CUOCERE
V.GERU	V.MC_20	N;I(w);N	2	CUOCERE
V.IND.PRES.1S	V.IND.PRES.3P	N;D(no)	1	ESSERE
V.IND.PRES.1S	V.IND.PRES.3P	N;D(on);N	1	ESSERE
V.IND.PRES.1S	V.IND.PRES.3P	N;S(do,n)	1	ANDARE
V.IND.PRES.1S	V.IND.PRES.3P	N;S(bb,v);N	1	DOVERE
V.IND.PRES.1S	V.IND.PRES.3P	N;S(ttʃo,n)	1	FARE
V.IND.PRES.1S	V.IND.PRES.3P	N;S(v,bb);N	1	DOVERE
V.IND.PRES.1S	V.MC_15	N;S(bbo,va)	1	DOVERE
V.IND.PRES.1S	V.MC_15	N;S(ono,ia)	1	ESSERE
V.IND.PRES.1S	V.MC_15	N;S(vo,bba)	1	DOVERE
V.IND.PRES.1S	V.MC_15	N;S(ɔ,abbja)	1	AVERE
V.IND.PRES.1S	V.MC_15	N;S(ɔ,appja)	1	SAPERE
V.IND.PRES.2S	V.IND.PRES.3S	N;S(i,je)	1	COMPIERE
V.IND.PRES.2S	V.IND.PRES.3S	N;S(i,le)	1	VOLERE
V.IND.PRES.2S	V.IND.PRES.3S	D(s);N;D(j)	1	ESSERE
V.IND.PRES.2S	V.IMP.2S	N;D(tʃi)	1	DIRE
V.IND.PRES.2S	V.IMP.2S	N;D(itʃ);N	1	DIRE
V.IND.PRES.2S	V.IMP.2S	N;S(j,bbi)	1	AVERE
V.IND.PRES.2S	V.IMP.2S	N;S(j,ppi)	1	SAPERE
V.IND.PRES.2S	V.IMP.2S	N;S(ɛ,i);N	1	ESSERE
V.IND.PRES.2S	V.IMP.2S	N;D(w);N;I(ʎʎ);N	1	VOLERE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.IND.PRES.3S	V.IMP.2S	N;D(tʃe)	1	DIRE
V.IND.PRES.3S	V.IMP.2S	N;I(bbi)	1	AVERE
V.IND.PRES.3S	V.IMP.2S	N;I(ppi)	1	SAPERE
V.IND.PRES.3S	V.IMP.2S	N;S(je,i)	1	COMPIERE
V.IND.PRES.3S	V.IMP.2S	I(s);N;S(ɛ,ij)	1	ESSERE
V.IND.PRES.3S	V.IMP.2S	N;D(w);N;S(le,ʎʎi)	1	VOLERE
V.MC_4	V.GERU	N;S(j,r'e)	1	PARERE
V.MC_4	V.GERU	N;S(pj,'e)	1	SAPERE
V.MC_4	V.GERU	N;S(tʃ,ʃ'e)	1	FARE
V.MC_4	V.GERU	N;D(t);N;I('e)	1	FARE
V.MC_4	V.GERU	N;D(w);N;I('e)	1	CUOCERE
V.MC_4	V.GERU	N;I(w);N;I('e)	1	CUOCERE
V.MC_4	V.GERU	I(e);N;S(j,s'ɛ)	1	ESSERE
V.MC_4	V.GERU	I(es);N;S(j,'ɛ)	1	ESSERE
V.MC_4	V.GERU	N;D(w);N;S(j,'e)	1	MUOVERE
V.MC_4	V.GERU	N;I(w);N;S(j,'e)	1	MUOVERE
V.MC_4	V.MC_20	N;S(j,r'e)	1	PARERE
V.MC_4	V.MC_20	N;S(pj,j'e)	1	SAPERE
V.MC_4	V.MC_20	N;S(tʃ,ʃ'e)	1	FARE
V.MC_4	V.MC_20	N;S(bbj,v'e)	1	AVERE
V.MC_4	V.MC_20	N;D(p);N;I('e)	1	SAPERE
V.MC_4	V.MC_20	N;D(t);N;I('e)	1	FARE
V.MC_4	V.MC_20	N;D(w);N;I('e)	1	CUOCERE
V.MC_4	V.MC_20	N;I(w);N;I('e)	1	CUOCERE
V.MC_4	V.MC_20	I(e);N;S(j,s'ɛ)	1	ESSERE
V.MC_4	V.MC_20	I(es);N;S(j,'ɛ)	1	ESSERE
V.MC_4	V.MC_20	N;D(w);N;S(j,'e)	1	MUOVERE
V.MC_4	V.MC_20	N;I(w);N;S(j,'e)	1	MUOVERE
V.IND.PRES.2P	V.MC_7	D(sj);N;I(r)	1	ESSERE
V.IND.PRES.2P	V.MC_7	N;I(atʃ);N;S(a,ev)	1	FARE
V.IND.PRES.2P	V.MC_7	N;I(itʃ);N;S(i,ev)	1	DIRE
V.IND.PRES.2P	V.MC_8	S(sj'ɛ,era)	1	ESSERE
V.IND.PRES.2P	V.MC_8	N;S('a,atʃe)	1	FARE
V.IND.PRES.2P	V.MC_8	N;S('i,itʃe)	1	DIRE
V.IND.PRES.2P	V.MC_14	N;D('a)	1	ANDARE
V.IND.PRES.2P	V.MC_14	N;S('e,r)	1	TRARRE
V.IND.PRES.2P	V.MC_14	N;S(j'ɛ,a)	1	ESSERE
V.IND.PRES.2P	V.MC_14	N;S(n'i,r)	1	VENIRE
V.IND.PRES.2P	V.MC_14	N;S(v'e,r)	1	BERE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.IND.PRES.2P	V.IMP.2P	N;S(ɛ,a)	1	ESSERE
V.IND.PRES.2P	V.IMP.2P	N;I(pj);N;S(e,a)	1	SAPERE
V.IND.PRES.2P	V.IMP.2P	N;S(l,ʎʎ);N;S(e,a)	1	VOLERE
V.IND.PRES.2P	V.IMP.2P	N;S(v,bbj);N;S(e,a)	1	AVERE
V.IND.PRES.2P	V.IMP.2P	N;I(p);N;I(j);N;S(e,a)	1	SAPERE
V.IND.PRES.2P	V.INF	N;D(ev);N	1	BERE
V.IND.PRES.2P	V.INF	N;S(a'e,'ar)	1	TRARRE
V.IND.PRES.2P	V.INF	I(');N;D(');N	1	ARDERE
V.IND.PRES.2P	V.INF	N;S(on'e,'or)	1	PORRE
V.IND.PRES.2P	V.INF	S(sj'ɛ,'ɛsse)	1	ESSERE
V.IND.PRES.2P	V.INF	D(sj);N;I(sse)	1	ESSERE
V.IND.PRES.2P	V.INF	N;S(od','ɔd);N	1	RODERE
V.IND.PRES.2P	V.INF	N;S(ov','ɔv);N	1	MUOVERE
V.IND.PRES.2P	V.INF	N;S(ov',w'ɔv);N	1	MUOVERE
V.IND.PRES.2P	V.INF	N;D(a);N;S(e,ar)	1	TRARRE
V.IND.PRES.2P	V.INF	N;I(');N;S('e,r)	1	TRARRE
V.IND.PRES.2P	V.INF	I('ɛ);N;S(j'ɛ,se)	1	ESSERE
V.IND.PRES.2P	V.INF	N;D(on);N;S(e,or)	1	PORRE
V.IND.PRES.2P	V.INF	N;I(');N;S('i,je)	1	COMPIERE
V.IND.PRES.2P	V.INF	N;I(');N;S(n'e,r)	1	PORRE
V.IND.PRES.2P	V.INF	N;S(ov,w);N;I(ɔv);N	1	MUOVERE
V.IND.PRES.2P	V.INF	N;S(el,'ɛ);N;S(',l);N	1	ESPELLERE
V.IND.PRES.3P	V.MC_15	N;S(n,da)	1	ANDARE
V.IND.PRES.3P	V.MC_15	N;S(o,ia)	1	ESSERE
V.IND.PRES.3P	V.MC_15	N;S(bbo,va)	1	DOVERE
V.IND.PRES.3P	V.MC_15	N;S(n,bbja)	1	AVERE
V.IND.PRES.3P	V.MC_15	N;S(n,ppja)	1	SAPERE
V.IND.PRES.3P	V.MC_15	N;S(n,ttʃa)	1	FARE
V.IND.PRES.3P	V.MC_15	N;S(vo,bba)	1	DOVERE
V.MC_7	V.MC_8	S('ɛr,era)	1	ESSERE
V.MC_7	V.MC_8	S('ɛ,e);N;I(a)	1	ESSERE
V.MC_7	V.MC_14	N;D('av)	1	ANDARE
V.MC_7	V.MC_14	S('ɛr,sa)	1	ESSERE
V.MC_7	V.MC_14	N;S('ev,r)	1	TRARRE
V.MC_7	V.MC_14	N;S(n'iv,r)	1	VENIRE
V.MC_7	V.MC_14	N;S(v'ev,r)	1	BERE
V.MC_7	V.MC_14	N;D(en');N;S(v,r)	1	TENERE
V.MC_7	V.MC_14	N;D(ev');N;S(v,r)	1	BERE
V.MC_7	V.IMP.2P	S('ɛr,sj'a)	1	ESSERE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.MC_7	V.IMP.2P	N;S('ev,pj'a)	1	SAPERE
V.MC_7	V.IMP.2P	N;S(l'ev,ʎʎ'a)	1	VOLERE
V.MC_7	V.IMP.2P	I(sj);N;S(ɛr,a)	1	ESSERE
V.MC_7	V.IMP.2P	N;S(v'ev,bbj'a)	1	AVERE
V.MC_7	V.IMP.2P	N;I(pj);N;S(ev,a)	1	SAPERE
V.MC_7	V.IMP.2P	N;D(atʃ);N;S(ev,a)	1	FARE
V.MC_7	V.IMP.2P	N;D(itʃ);N;S(ev,i)	1	DIRE
V.MC_7	V.IMP.2P	N;I(p);N;S('ev,j'a)	1	SAPERE
V.MC_7	V.IMP.2P	N;S(l,ʎʎ);N;S(ev,a)	1	VOLERE
V.MC_7	V.IMP.2P	N;S(v,bbj);N;S(ev,a)	1	AVERE
V.MC_7	V.IMP.2P	N;I(p);N;I(j);N;S(ev,a)	1	SAPERE
V.MC_7	V.INF	N;S(r,sse)	1	ESSERE
V.MC_7	V.INF	N;D(ev);N;D(v)	1	BERE
V.MC_7	V.INF	N;D(ev);N;I(e)	1	RICEVERE
V.MC_7	V.INF	N;D(a);N;S(ev,ar)	1	TRARRE
V.MC_7	V.INF	I(');N;D(');N;D(v)	1	ARDERE
V.MC_7	V.INF	N;D(atʃ);N;S(ev,a)	1	FARE
V.MC_7	V.INF	N;D(itʃ);N;S(ev,i)	1	DIRE
V.MC_7	V.INF	N;D(on);N;S(ev,or)	1	PORRE
V.MC_7	V.INF	N;I(');N;S('iv,je)	1	COMPIERE
V.MC_7	V.INF	N;S(od','ɔd);N;D(v)	1	RODERE
V.MC_7	V.INF	N;S(ov','ɔv);N;D(v)	1	MUOVERE
V.MC_7	V.INF	N;D(ov);N;S(e,ɔ);N;I(e)	1	MUOVERE
V.MC_7	V.INF	N;S(o,w'ɔ);N;D(');N;D(v)	1	CUOCERE
V.MC_7	V.INF	N;S(ov,w);N;S(e,ɔ);N;I(e)	1	MUOVERE
V.MC_7	V.INF	N;S(el,'ɛ);N;S(',l);N;D(v)	1	ESPELLERE
V.MC_8	V.MC_14	N;D(a)	1	ANDARE
V.MC_8	V.MC_14	N;S(e,r)	1	TRARRE
V.MC_8	V.MC_14	N;S(ni,r)	1	VENIRE
V.MC_8	V.MC_14	N;S(ve,r)	1	BERE
V.MC_8	V.MC_14	S(er,s);N	1	ESSERE
V.MC_8	V.IMP.2P	N;S(e,pj'a)	1	SAPERE
V.MC_8	V.IMP.2P	S(er,sj');N	1	ESSERE
V.MC_8	V.IMP.2P	N;S(atʃe,'a)	1	FARE
V.MC_8	V.IMP.2P	N;S(itʃe,'i)	1	DIRE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.MC_8	V.IMP.2P	N;S(le,ʎʎ'a)	1	VOLERE
V.MC_8	V.IMP.2P	N;S(ve,bbj'a)	1	AVERE
V.MC_8	V.IMP.2P	N;I(p);N;S(e,j'a)	1	SAPERE
V.MC_8	V.INF	I(');N	1	ARDERE
V.MC_8	V.INF	N;S(ev,');N	1	BERE
V.MC_8	V.INF	N;S(atʃe,'a)	1	FARE
V.MC_8	V.INF	N;S(itʃe,'i)	1	DIRE
V.MC_8	V.INF	N;S(one,'or)	1	PORRE
V.MC_8	V.INF	S(era,'ɛsse)	1	ESSERE
V.MC_8	V.INF	N;I(');N;S(e,r)	1	TRARRE
V.MC_8	V.INF	N;I(');N;S(i,je)	1	COMPIERE
V.MC_8	V.INF	N;I(');N;S(ne,r)	1	PORRE
V.IND.PREM.1S	V.MC_21	N;S(di,st)	1	VEDERE
V.IND.PREM.1S	V.MC_21	N;S(ej,it)	1	ESISTERE
V.IND.PREM.1S	V.MC_21	N;S(zi,st)	1	RIMANERE
V.IND.PREM.1S	V.MC_21	N;S(kkwɪ,t)	1	NASCERE
V.IND.PREM.1S	V.MC_21	N;S(ezi,ɛst)	1	CHIEDERE
V.IND.PREM.1S	V.MC_21	N;S(insi,ett)	1	STRINGERE
V.IND.PREM.1S	V.MC_21	N;S(issi,ett)	1	DIRE
V.IND.PREM.1S	V.MC_21	N;S(uppi,ott)	1	ROMPERE
V.IND.PREM.1S	V.MC_21	N;S(ɔssi,ott)	1	CUOCERE
V.IND.PREM.1S	V.MC_21	N;S(r'ij,'ɛrt)	1	APRIRE
V.IND.PREM.1S	V.MC_21	S('ebbi,av'ut)	1	AVERE
V.IND.PREM.1S	V.MC_21	N;S(o,ɔ);N;D(i)	1	RODERE
V.IND.PREM.1S	V.MC_21	N;S(or'ij,'ɔrt)	1	MORIRE
V.IND.PREM.1S	V.MC_21	N;S('eppi,ap'ut)	1	SAPERE
V.IND.PREM.1S	V.MC_21	N;S('ɔlli,ol'ut)	1	VOLERE
V.IND.PREM.1S	V.MC_21	N;S(e,a);N;D(ti)	1	STARE
V.IND.PREM.1S	V.MC_21	N;S(et,a);N;D(i)	1	STARE
V.IND.PREM.1S	V.MC_21	N;S(idʒ'ej,'att)	1	ESIGERE
V.IND.PREM.1S	V.MC_21	N;S(ol'ej,'ɔlit)	1	SOLERE
V.IND.PREM.1S	V.MC_21	N;D(');N;S(i,'ut)	1	VIVERE
V.IND.PREM.1S	V.MC_21	N;D(');N;S(si,'s)	1	VALERE
V.IND.PREM.1S	V.MC_21	N;D(v);N;S(ej,ut)	1	DEVOLVERE
V.IND.PREM.1S	V.MC_21	N;I(j);N;S(ij,ut)	1	COMPIERE
V.IND.PREM.1S	V.MC_21	N;S('ebbi,eʃʃ'ut)	1	CRESCERE
V.IND.PREM.1S	V.MC_21	N;S('obbi,oʃʃ'ut)	1	CONOSCERE
V.IND.PREM.1S	V.MC_21	N;S('ɛrsi,erd'ut)	1	PERDERE
V.IND.PREM.1S	V.MC_21	N;D(');N;S(di,'ut)	1	CADERE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.IND.PREM.1S	V.MC_21	N;D(');N;S(si,'ut)	1	DOLERE
V.IND.PREM.1S	V.MC_21	N;D(');N;S(vi,'ut)	1	BERE
V.IND.PREM.1S	V.MC_21	N;D(ett);N;S(j,ss)	1	CONNETTERE
V.IND.PREM.1S	V.MC_21	N;D(j);N;S(edi,at)	1	DARE
V.IND.PREM.1S	V.MC_21	N;D(r);N;S(ij,ɛrt)	1	APRIRE
V.IND.PREM.1S	V.MC_21	N;S(e,a);N;S(ʃi,t)	1	FARE
V.IND.PREM.1S	V.MC_21	S(f,st);N;S(uj,at)	1	ESSERE
V.IND.PREM.1S	V.MC_21	N;D(');N;S(ssi,'tt)	1	TRARRE
V.IND.PREM.1S	V.MC_21	N;D(or);N;S(ij,ɔrt)	1	MORIRE
V.IND.PREM.1S	V.MC_21	N;D(');N;I(');N;D(i)	1	VALERE
V.IND.PREM.1S	V.MC_21	N;D(idʒ);N;S(ej,att)	1	ESIGERE
V.IND.PREM.1S	V.MC_21	N;S(o,'ɔ);N;S('ij,t)	1	MORIRE
V.IND.PREM.1S	V.MC_21	N;S('e,a);N;S(pi,'ut)	1	SAPERE
V.IND.PREM.1S	V.MC_21	N;S('ɔ,o);N;S(li,'ut)	1	VOLERE
V.IND.PREM.1S	V.MC_21	N;S(o,'ɔ);N;S('ej,it)	1	SOLERE
V.IND.PREM.1S	V.MC_21	N;S('ɛ,e);N;S(si,d'ut)	1	PERDERE
V.IND.PREM.1S	V.MC_21	N;S(e,i);N;S(insi,ett)	1	RESTRINGERE
V.IND.PREM.1S	V.MC_21	N;D(v);N;S(e,u);N;D(ti)	1	DEVOLVERE
V.IND.PREM.1S	V.MC_21	N;D(v);N;S(et,u);N;D(i)	1	DEVOLVERE
V.MC_10	V.IND.PREM.1P	N;S(o,u)	1	ESSERE
V.IND.PREM.3S	V.MC_21	N;S(e,it)	1	ESISTERE
V.IND.PREM.3S	V.MC_21	N;S(de,st)	1	VEDERE
V.IND.PREM.3S	V.MC_21	N;S(ze,st)	1	RIMANERE
V.IND.PREM.3S	V.MC_21	N;S(kkwe,t)	1	NASCERE
V.IND.PREM.3S	V.MC_21	N;S(eze,ɛst)	1	CHIEDERE
V.IND.PREM.3S	V.MC_21	N;S(v'e,'ut)	1	DEVOLVERE
V.IND.PREM.3S	V.MC_21	N;S(inse,ett)	1	STRINGERE
V.IND.PREM.3S	V.MC_21	N;S(isse,ett)	1	DIRE
V.IND.PREM.3S	V.MC_21	N;S(r'i,'ɛrt)	1	APRIRE
V.IND.PREM.3S	V.MC_21	N;S(uppe,ott)	1	ROMPERE
V.IND.PREM.3S	V.MC_21	N;S(ɔsse,ott)	1	CUOCERE
V.IND.PREM.3S	V.MC_21	N;S(or'i,'ɔrt)	1	MORIRE
V.IND.PREM.3S	V.MC_21	S('ebbe,av'ut)	1	AVERE
V.IND.PREM.3S	V.MC_21	N;S(ett'e,'ess)	1	CONNETTERE
V.IND.PREM.3S	V.MC_21	N;S(idʒ'e,'att)	1	ESIGERE
V.IND.PREM.3S	V.MC_21	N;S(o,ɔ);N;D(e)	1	RODERE
V.IND.PREM.3S	V.MC_21	N;D(ett);N;I(ss)	1	CONNETTERE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.IND.PREM.3S	V.MC_21	N;D(v);N;S(e,ut)	1	DEVOLVERE
V.IND.PREM.3S	V.MC_21	N;I(j);N;S(i,ut)	1	COMPIERE
V.IND.PREM.3S	V.MC_21	N;S('eppe,ap'ut)	1	SAPERE
V.IND.PREM.3S	V.MC_21	N;S('ɔlle,ol'ut)	1	VOLERE
V.IND.PREM.3S	V.MC_21	N;S(e,a);N;D(te)	1	STARE
V.IND.PREM.3S	V.MC_21	N;S(et,a);N;D(e)	1	STARE
V.IND.PREM.3S	V.MC_21	N;D(');N;S(e,'ut)	1	VIVERE
V.IND.PREM.3S	V.MC_21	N;D(');N;S(se,'s)	1	VALERE
V.IND.PREM.3S	V.MC_21	N;D(r);N;S(i,ɛrt)	1	APRIRE
V.IND.PREM.3S	V.MC_21	N;I('ɛ);N;S('i,t)	1	APRIRE
V.IND.PREM.3S	V.MC_21	N;S('ebbe,eʃʃ'ut)	1	CRESCERE
V.IND.PREM.3S	V.MC_21	N;S('obbe,oʃʃ'ut)	1	CONOSCERE
V.IND.PREM.3S	V.MC_21	N;S('ɛrse,erd'ut)	1	PERDERE
V.IND.PREM.3S	V.MC_21	S(f,st);N;S(u,at)	1	ESSERE
V.IND.PREM.3S	V.MC_21	N;D(');N;S(de,'ut)	1	CADERE
V.IND.PREM.3S	V.MC_21	N;D(');N;S(se,'ut)	1	DOLERE
V.IND.PREM.3S	V.MC_21	N;D(');N;S(ve,'ut)	1	BERE
V.IND.PREM.3S	V.MC_21	N;D(j);N;S(ede,at)	1	DARE
V.IND.PREM.3S	V.MC_21	N;S(e,a);N;S(ʃe,t)	1	FARE
V.IND.PREM.3S	V.MC_21	N;D(');N;S(sse,'tt)	1	TRARRE
V.IND.PREM.3S	V.MC_21	N;I(');N;S(tt'e,ss)	1	CONNETTERE
V.IND.PREM.3S	V.MC_21	N;S(o,'ɔ);N;S('i,t)	1	MORIRE
V.IND.PREM.3S	V.MC_21	N;D(');N;I(');N;D(e)	1	VALERE
V.IND.PREM.3S	V.MC_21	N;S(o,'ɔ);N;S('e,it)	1	SOLERE
V.IND.PREM.3S	V.MC_21	N;S('e,a);N;S(pe,'ut)	1	SAPERE
V.IND.PREM.3S	V.MC_21	N;S('ɔ,o);N;S(le,'ut)	1	VOLERE
V.IND.PREM.3S	V.MC_21	N;S('ɛ,e);N;S(se,d'ut)	1	PERDERE
V.IND.PREM.3S	V.MC_21	N;S(e,i);N;S(inse,ett)	1	RESTRINGERE
V.IND.PREM.3S	V.MC_21	N;D(v);N;S(e,u);N;D(te)	1	DEVOLVERE
V.IND.PREM.3S	V.MC_21	N;D(v);N;S(et,u);N;D(e)	1	DEVOLVERE
V.IND.PREM.3P	V.MC_21	N;S(der,st)	1	VEDERE
V.IND.PREM.3P	V.MC_21	N;S(zer,st)	1	RIMANERE
V.IND.PREM.3P	V.MC_21	N;S(eron,it)	1	ESISTERE
V.IND.PREM.3P	V.MC_21	N;S(kkwer,t)	1	NASCERE
V.IND.PREM.3P	V.MC_21	N;S(ezer,ɛst)	1	CHIEDERE
V.IND.PREM.3P	V.MC_21	N;S(inser,ett)	1	STRINGERE
V.IND.PREM.3P	V.MC_21	N;S(isser,ett)	1	DIRE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.IND.PREM.3P	V.MC_21	N;S(upper,ott)	1	ROMPERE
V.IND.PREM.3P	V.MC_21	N;S(ɔsser,ott)	1	CUOCERE
V.IND.PREM.3P	V.MC_21	N;S('iron,j'ut)	1	COMPIERE
V.IND.PREM.3P	V.MC_21	S('ebber,av'ut)	1	AVERE
V.IND.PREM.3P	V.MC_21	S(f'uron,st'at)	1	ESSERE
V.IND.PREM.3P	V.MC_21	N;S(o,ɔ);N;D(er)	1	RODERE
V.IND.PREM.3P	V.MC_21	N;S(e,a);N;D(ter)	1	STARE
V.IND.PREM.3P	V.MC_21	N;S(et,a);N;D(er)	1	STARE
V.IND.PREM.3P	V.MC_21	N;D(');N;S(er,'ut)	1	VIVERE
V.IND.PREM.3P	V.MC_21	N;D(');N;S(ser,'s)	1	VALERE
V.IND.PREM.3P	V.MC_21	N;D(ins);N;S(r,tt)	1	STRINGERE
V.IND.PREM.3P	V.MC_21	N;D(iss);N;S(r,tt)	1	DIRE
V.IND.PREM.3P	V.MC_21	N;S('ebber,eʃʃ'ut)	1	CRESCERE
V.IND.PREM.3P	V.MC_21	N;S('obber,oʃʃ'ut)	1	CONOSCERE
V.IND.PREM.3P	V.MC_21	N;S('ɛrser,erd'ut)	1	PERDERE
V.IND.PREM.3P	V.MC_21	N;D(');N;S(der,'ut)	1	CADERE
V.IND.PREM.3P	V.MC_21	N;D(');N;S(ser,'ut)	1	DOLERE
V.IND.PREM.3P	V.MC_21	N;D(');N;S(ver,'ut)	1	BERE
V.IND.PREM.3P	V.MC_21	N;D(j);N;S(eder,at)	1	DARE
V.IND.PREM.3P	V.MC_21	N;D(v);N;S(eron,ut)	1	DEVOLVERE
V.IND.PREM.3P	V.MC_21	N;I(j);N;S(iron,ut)	1	COMPIERE
V.IND.PREM.3P	V.MC_21	N;S(e,a);N;S(ʃer,t)	1	FARE
V.IND.PREM.3P	V.MC_21	N;D(');N;S(sser,'tt)	1	TRARRE
V.IND.PREM.3P	V.MC_21	N;D(ett);N;S(ron,ss)	1	CONNETTERE
V.IND.PREM.3P	V.MC_21	S(f,st);N;S(uron,at)	1	ESSERE
V.IND.PREM.3P	V.MC_21	N;D(');N;I(');N;D(er)	1	VALERE
V.IND.PREM.3P	V.MC_21	N;D(idʒ);N;S(eron,att)	1	ESIGERE
V.IND.PREM.3P	V.MC_21	N;D(ol);N;S(eron,ɔlit)	1	SOLERE
V.IND.PREM.3P	V.MC_21	N;S('e,a);N;S(per,'ut)	1	SAPERE
V.IND.PREM.3P	V.MC_21	N;S('ɔ,o);N;S(ler,'ut)	1	VOLERE
V.IND.PREM.3P	V.MC_21	N;S('ɛ,e);N;S(ser,d'ut)	1	PERDERE
V.IND.PREM.3P	V.MC_21	N;S(e,i);N;S(inser,ett)	1	RESTRINGERE
V.IND.PREM.3P	V.MC_21	N;D(v);N;S(e,u);N;D(te r)	1	DEVOLVERE
V.IND.PREM.3P	V.MC_21	N;D(v);N;S(et,u);N;D(e r)	1	DEVOLVERE
V.IND.PREM.3P	V.MC_21	N;D(r);N;S(i,ɛ);N;S(on, t)	1	APRIRE
V.IND.PREM.3P	V.MC_21	N;D(or);N;S(i,ɔ);N;S(o	1	MORIRE

form A	*formB*	*relation A→B*	*lexemes*	*example*
		n,t)		
V.IND.PREM.3P	V.MC_21	N;S(e,i);N;D(ins);N;S(r ,tt)	1	RESTRINGERE
V.MC_14	V.IMP.2P	N;I('a)	1	ANDARE
V.MC_14	V.IMP.2P	N;I(j');N	1	ESSERE
V.MC_14	V.IMP.2P	N;I(pj'a)	1	SAPERE
V.MC_14	V.IMP.2P	N;S(r,'e)	1	TRARRE
V.MC_14	V.IMP.2P	N;S(r,n'i)	1	VENIRE
V.MC_14	V.IMP.2P	N;S(r,v'e)	1	BERE
V.MC_14	V.IMP.2P	N;S(r,ʎʎ'a)	1	VOLERE
V.MC_14	V.IMP.2P	N;S(v,bbj'a)	1	AVERE
V.MC_14	V.IMP.2P	N;I(p);N;I(j'a)	1	SAPERE
V.MC_14	V.INF	I(');N	1	ARDERE
V.MC_14	V.INF	N;I('a)	1	ANDARE
V.MC_14	V.INF	N;S(er,'e)	1	BERE
V.MC_14	V.INF	N;S(r,n'i)	1	VENIRE
V.MC_14	V.INF	N;I(');N;D(r)	1	BERE
V.MC_14	V.INF	N;I(');N;I(e)	1	VIVERE
V.MC_14	V.INF	I('ɛ);N;S(a,se)	1	ESSERE
V.MC_14	V.INF	I('ɛs);N;S(a,e)	1	ESSERE
V.MC_14	V.INF	N;I(');N;S(i,je)	1	COMPIERE
V.IMP.2P	V.INF	N;D(ev);N	1	BERE
V.IMP.2P	V.INF	N;S(a'e,'ar)	1	TRARRE
V.IMP.2P	V.INF	I(');N;D(');N	1	ARDERE
V.IMP.2P	V.INF	N;S(on'e,'or)	1	PORRE
V.IMP.2P	V.INF	S(sj'a,'ɛsse)	1	ESSERE
V.IMP.2P	V.INF	N;S(od','ɔd);N	1	RODERE
V.IMP.2P	V.INF	N;S(ov','ɔv);N	1	MUOVERE
V.IMP.2P	V.INF	N;S(ov',w'ɔv);N	1	MUOVERE
V.IMP.2P	V.INF	N;D(a);N;S(e,ar)	1	TRARRE
V.IMP.2P	V.INF	N;D(pj);N;S(a,e)	1	SAPERE
V.IMP.2P	V.INF	N;I(');N;S('e,r)	1	TRARRE
V.IMP.2P	V.INF	I('ɛ);N;S(j'a,se)	1	ESSERE
V.IMP.2P	V.INF	N;D(on);N;S(e,or)	1	PORRE
V.IMP.2P	V.INF	N;I(');N;S('i,je)	1	COMPIERE
V.IMP.2P	V.INF	N;I(');N;S(n'e,r)	1	PORRE
V.IMP.2P	V.INF	N;S(ʎʎ,l);N;S(a,e)	1	VOLERE
V.IMP.2P	V.INF	N;S(bbj,v);N;S(a,e)	1	AVERE
V.IMP.2P	V.INF	N;S(ov,w);N;I(ɔv);N	1	MUOVERE

form A	*formB*	*relation A→B*	*lexemes*	*example*
V.IMP.2P	V.INF	N;S(el,'ɛ);N;S(',l);N	1	ESPELLERE
V.IMP.2P	V.INF	N;D(p);N;D(j);N;S(a,e)	1	SAPERE
V.GERU	V.MC_20	N;I(j);N	1	SAPERE

Table 15: all the relations between metacells.

3.11 A CLASSIFICATION OF LEXEMES

After the computation of relations between metacells for every lexeme, metacells having the smallest conditional entropy values (see § 2.3) are connected. The conditional entropy is computed on the set of lexemes, not weighed by frequency: the overwhelming predictability of verbs in *-are* is not represented. Interconnected metacells are thus grouped, as illustrated in Figure 5:

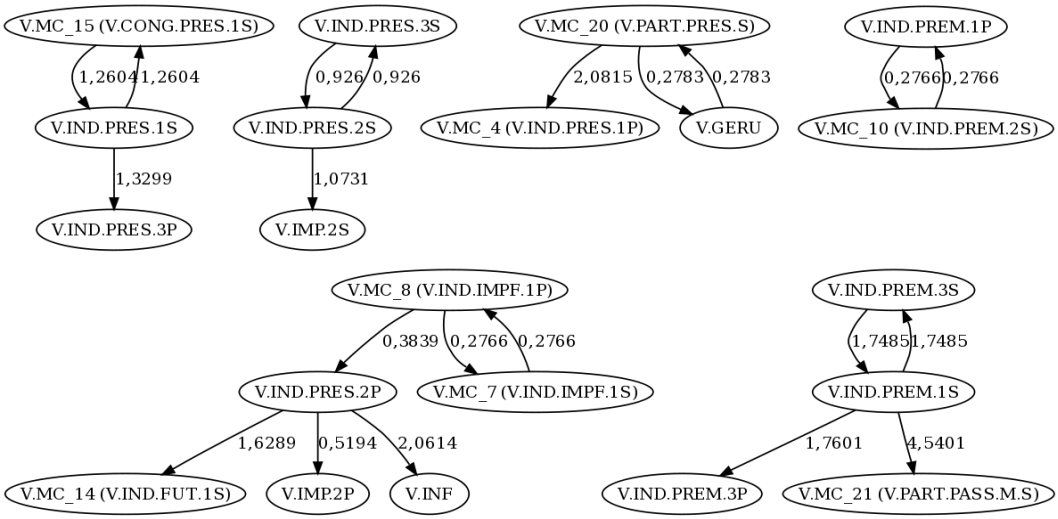


Figure 5: interconnection of metacells by minimum conditional entropy value.

Finally, the lexemes have been clustered according to set of relations between metacells they realise (§ 3.10). Figure 6 to Figure 11 show the

dendrograms of lexeme clustering, computed with an euclidean distance metric between lexemes on the set of relations between each pair of forms (or metacells) contained respectively in each grouping depicted in Figure 5 above.

One dendrogram has been computed for each metacell grouping. For each grouping, all the lexemes have been considered; and for each lexeme, a list of boolean values (1 or 0) has been built, corresponding to the list of the formal relations (as shown in Table 15), where "1" means that the specific relation is valid for the specific lexeme. The distance between each pair of elements of the dendrogram (lexemes) is the euclidean distance between their lists: in this case, the square root of the number of relations that are instanced by only one lexeme of the pair.

Defective and overabundant lexemes have not been included in this computation for clarity of illustration.

Figure 6: clustering of lexemes according to relations between Indicative Present 1s, Indicative Present 3p, Metacell 15 (Subjunctive Present 1s).

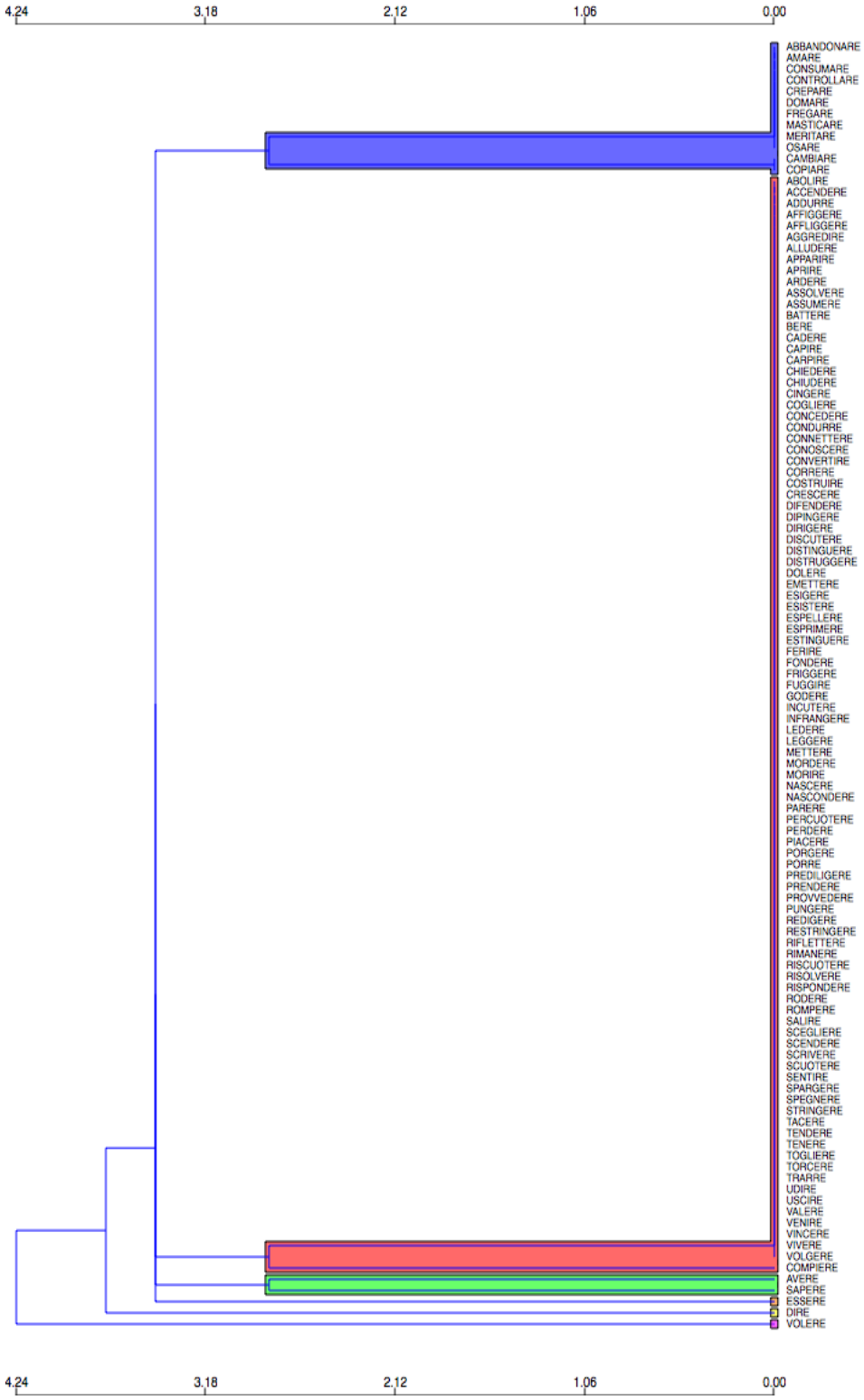


Figure 7: clustering of lexemes according to relations between Indicative Present 2s, Indicative Present 3s, Imperative 2s.

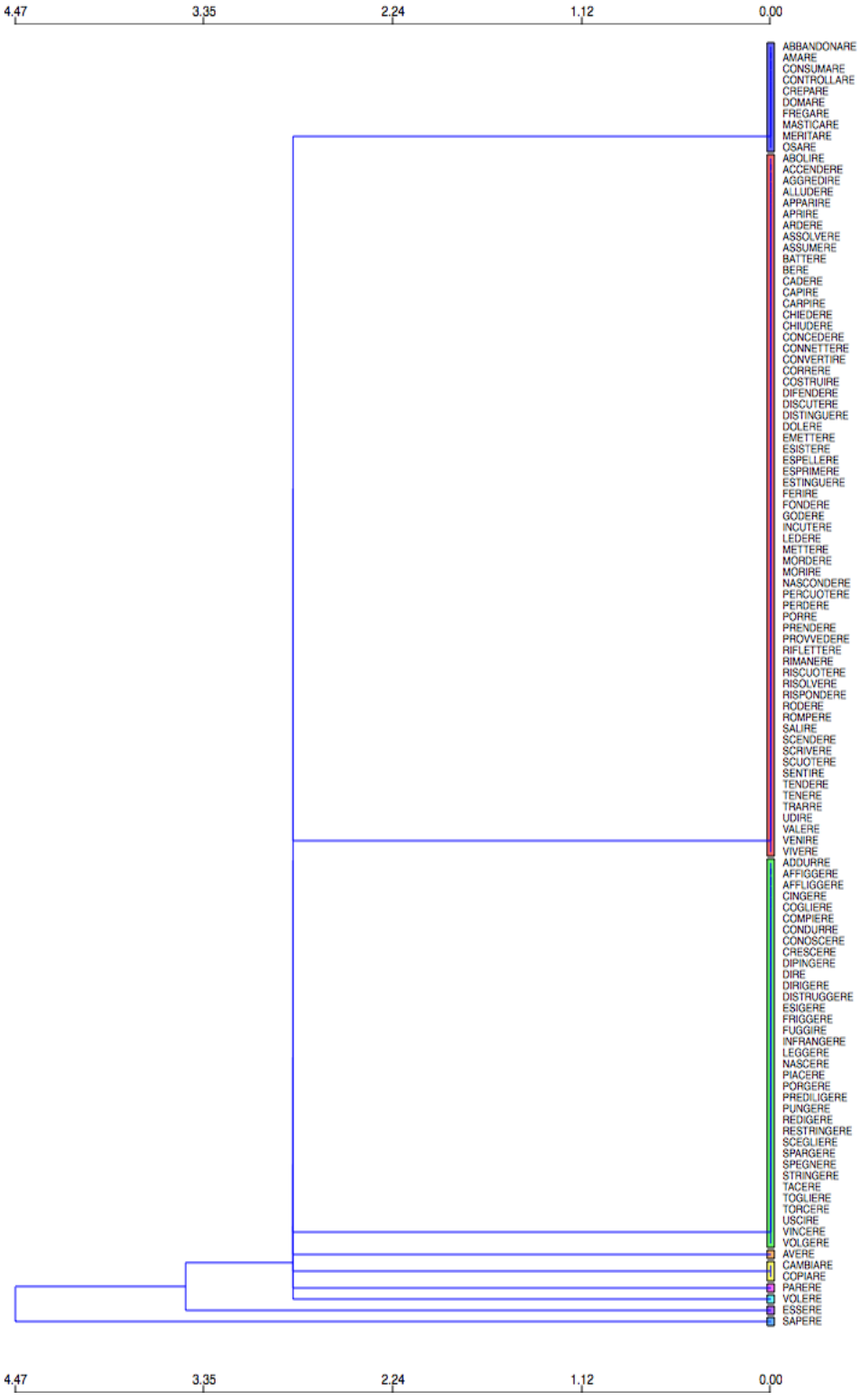


Figure 8: clustering of lexemes according to relations between Indicative Present 1p, Metacell 20 (Present Participle), Gerund.

Figure 9: clustering of lexemes according to relations between Simple Past 1p and Metacell 10 (Simple Past 2s).

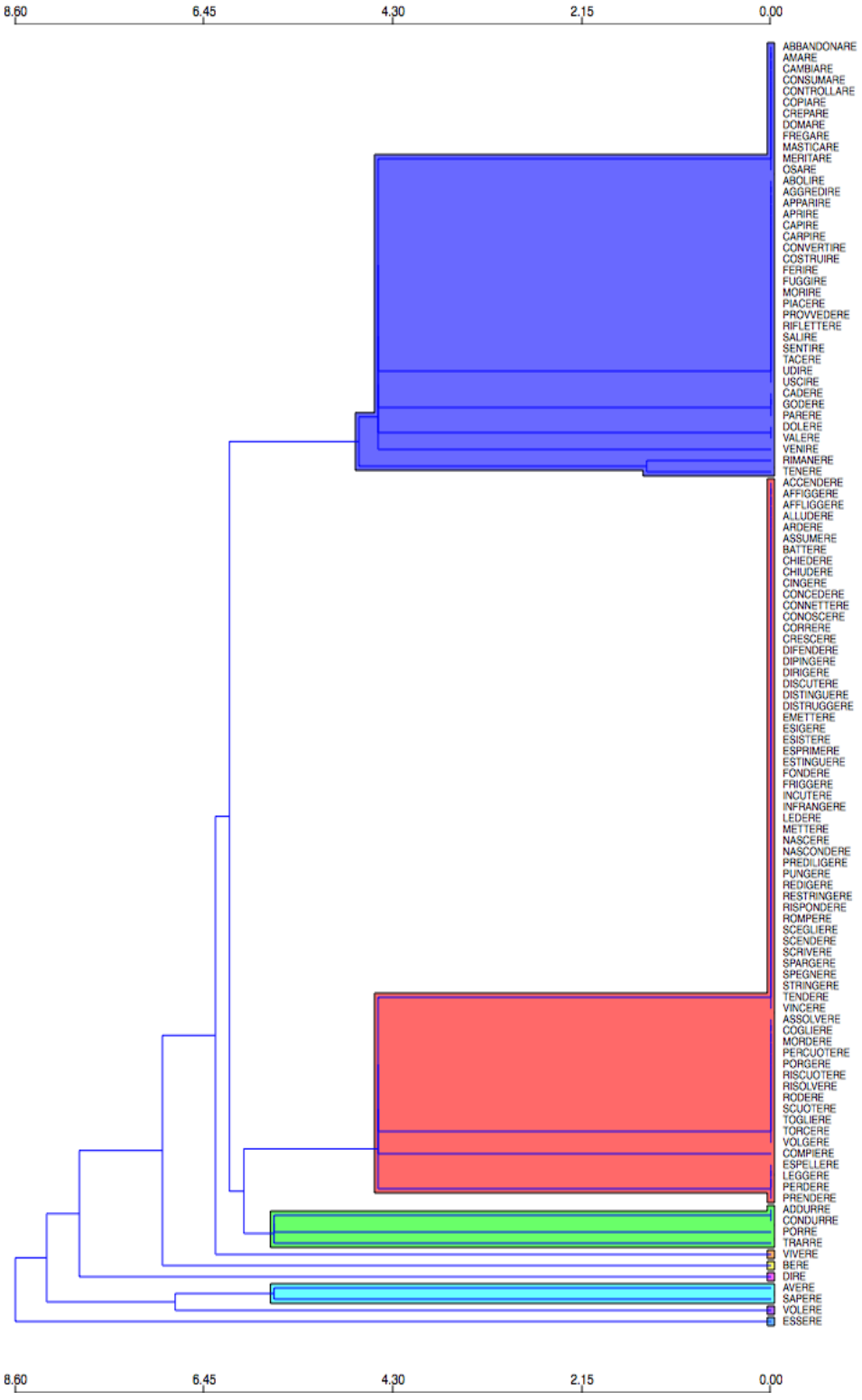


Figure 10: clustering of lexemes according to relations between Metacell 8 (Imperfect 1p), Metacell 7 (Imperfect 1s), Indicative Present 2p, Metacell 14 (Future 1s), Imperative 2p, Infinite.

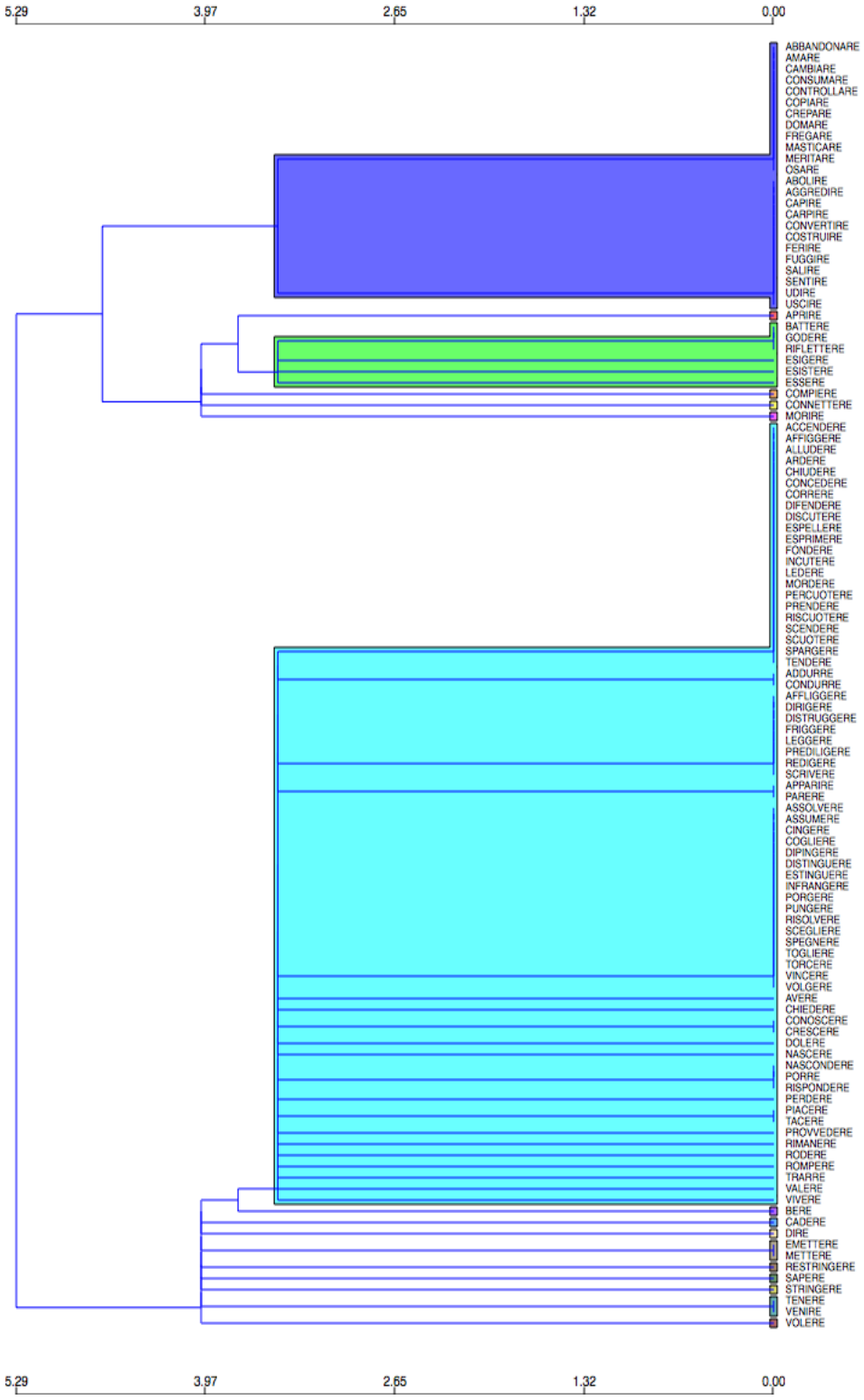


Figure 11: clustering of lexemes according to relations between Simple Past 3s, Simple Past 1s, Simple Past 3p, Metacell 21 (Past Participle).

4 Appendix: listing of computed stems

	ABBANDONARE	ABOLIRE	ACCENDERE	ADDURRE	AFFIGGERE	AFFLIGGERE
V.IND.PRES.1S	abband'ono	abol'isko	attʃ'endo	add'uko	aff'iggo	affl'iggo
V.IND.PRES.2S	abband'oni	abol'iʃʃi	attʃ'endi	add'utʃi	aff'iddʒi	affl'iddʒi
V.IND.PRES.3S	abband'ona	abol'iʃʃe	attʃ'ende	add'utʃe	aff'iddʒe	affl'iddʒe
V.MC_4 (V.IND.PRES.1P)	abbandonj	abolj	attʃendj	addutʃ	affiddʒ	afflidʒ
V.IND.PRES.2P	abbandon'a	abol'i	attʃend'e	addutʃ'e	affiddʒ'e	afflidʒ'e
V.IND.PRES.3P	abband'ona	abol'isko	attʃ'endo	add'uko	aff'iggo	affl'iggo
V.MC_7 (V.IND.IMPF.1S)	abbandon'av	abol'iv	attʃend'ev	addutʃ'ev	affiddʒ'ev	afflidʒ'ev
V.MC_8 (V.IND.IMPF.1P)	abbandona	aboli	attʃende	addutʃe	affiddʒe	afflidʒe
V.IND.PREM.1S	abbandon'aj	abol'ij	attʃ'ezi	add'ussi	aff'issi	affl'issi
V.MC_10 (V.IND.PREM.2S)	abbandon'a	abol'i	attʃend'e	addutʃ'e	affiddʒ'e	afflidʒ'e
V.IND.PREM.3S	abbandon'ɔ	abol'i	attʃ'eze	add'usse	aff'isse	affl'isse
V.IND.PREM.1P	abbandon'a	abol'i	attʃend'e	addutʃ'e	affiddʒ'e	afflidʒ'e
V.IND.PREM.3P	abbandon'aron	abol'iron	attʃ'ezer	add'usser	aff'isser	affl'isser
V.MC_14 (V.IND.FUT.1S)	abbandone	aboli	attʃende	addur	affiddʒe	afflidʒe
V.MC_15 (V.CONG.PRES.1S)	abband'oni	abol'iska	attʃ'enda	add'uka	aff'igga	affl'igga
V.IMP.2S	abband'ona	abol'iʃʃi	attʃ'endi	add'utʃi	aff'iddʒi	affl'iddʒi
V.IMP.2P	abbandon'a	abol'i	attʃend'e	addutʃ'e	affiddʒ'e	afflidʒ'e
V.GERU	abbandon'a	abol'e	attʃend'e	addutʃ'e	affiddʒ'e	afflidʒ'e
V.INF	abbandon'a	abol'i	attʃ'ende	add'ur	aff'iddʒe	affl'iddʒe
V.MC_20 (V.PART.PRES.S)	abbandon'a	abol'e	attʃend'e	addutʃ'e	affiddʒ'e	afflidʒ'e
V.MC_21 (V.PART.PASS.M.S)	abbandon'at	abol'it	attʃ'ez	add'ott	aff'iss	affl'itt

	AGGREDIRE	ALLUDERE	AMARE	ANDARE	APPARIRE	APRIRE
V.IND.PRES.1S	aggred'isko	all'udo	'amo	v'ado	app'ajo	'apro
V.IND.PRES.2S	aggred'iʃʃi	all'udi	'ami	v'aj	app'ari	'apri
V.IND.PRES.3S	aggred'iʃʃe	all'ude	'ama	v'a	app'are	'apre
V.MC_4 (V.IND.PRES.1P)	aggredj	alludj	amj	andj	apparj	aprj
V.IND.PRES.2P	aggred'i	allud'e	am'a	and'a	appar'i	apr'i
V.IND.PRES.3P	aggred'isko	all'udo	'ama	v'an	app'ajo	'apro
V.MC_7 (V.IND.IMPF.1S)	aggred'iv	allud'ev	am'av	and'av	appar'iv	apr'iv
V.MC_8 (V.IND.IMPF.1P)	aggredi	allude	ama	anda	appari	apri
V.IND.PREM.1S	aggred'ij	all'uzi	am'aj	and'aj	app'arvi	apr'ij
V.MC_10 (V.IND.PREM.2S)	aggred'i	allud'e	am'a	and'a	appar'i	apr'i
V.IND.PREM.3S	aggred'i	all'uze	am'ɔ	and'ɔ	app'arve	aprˈi
V.IND.PREM.1P	aggred'i	allud'e	am'a	and'a	appar'i	apr'i
V.IND.PREM.3P	aggred'iron	all'uzer	am'aron	and'aron	app'arver	apr'iron
V.MC_14 (V.IND.FUT.1S)	aggredi	allude	ame	and	appari	apri
V.MC_15 (V.CONG.PRES.1S)	aggred'iska	all'uda	'ami	v'ada	app'aja	'apra
V.IMP.2S	aggred'iʃʃi	all'udi	'ama	v'a v'aj	app'ari	'apri
V.IMP.2P	aggred'i	allud'e	am'a	and'a	appar'i	apr'i
V.GERU	aggred'e	allud'e	am'a	and'a	appar'e	apr'e
V.INF	aggred'i	all'ude	am'a	and'a	appar'i	aprˈi
V.MC_20 (V.PART.PRES.S)	aggred'e	allud'e	am'a	and'a	appar'e	apr'e
V.MC_21 (V.PART.PASS.M.S)	aggred'it	all'uz	am'at	and'at	app'ars	ap'ɛrt

	ARDERE	ASSOLVERE	ASSUMERE	AVERE	BATTERE	BERE
V.IND.PRES.1S	'ardo	ass'ɔlvo	ass'umo	'ɔ	b'atto	b'evo
V.IND.PRES.2S	'ardi	ass'ɔlvi	ass'umi	'aj	b'atti	b'evi
V.IND.PRES.3S	'arde	ass'ɔlve	ass'ume	'a	b'atte	b'eve
V.MC_4 (V.IND.PRES.1P)	ardj	assolvj	assumj	abbj	battj	bevj
V.IND.PRES.2P	ard'e	assolv'e	assum'e	av'e	batt'e	bev'e
V.IND.PRES.3P	'ardo	ass'ɔlvo	ass'umo	'an	b'atto	b'evo
V.MC_7 (V.IND.IMPF.1S)	ard'ev	assolv'ev	assum'ev	av'ev	batt'ev	bev'ev
V.MC_8 (V.IND.IMPF.1P)	arde	assolve	assume	ave	batte	beve
V.IND.PREM.1S	'arsi	ass'ɔlsi	ass'unsi	'ebbi	batt'ej	b'evvi
V.MC_10 (V.IND.PREM.2S)	ard'e	assolv'e	assum'e	av'e	batt'e	bev'e
V.IND.PREM.3S	'arse	ass'ɔlse	ass'unse	'ebbe	batt'e	b'evve
V.IND.PREM.1P	ard'e	assolv'e	assum'e	av'e	batt'e	bev'e
V.IND.PREM.3P	'arser	ass'ɔlser	ass'unser	'ebber	batt'eron	b'evver
V.MC_14 (V.IND.FUT.1S)	arde	assolve	assume	av	batte	ber
V.MC_15 (V.CONG.PRES.1S)	'arda	ass'ɔlva	ass'uma	'abbja	b'atta	b'eva
V.IMP.2S	'ardi	ass'ɔlvi	ass'umi	'abbi	b'atti	b'evi
V.IMP.2P	ard'e	assolv'e	assum'e	abbj'a	batt'e	bev'e
V.GERU	ard'e	assolv'e	assum'e	av'e	batt'e	bev'e
V.INF	'arde	ass'ɔlve	ass'ume	av'e	b'atte	b'e
V.MC_20 (V.PART.PRES.S)	ard'e	assolv'e	assum'e	av'e	batt'e	bev'e
V.MC_21 (V.PART.PASS.M.S)	'ars	ass'ɔlt	ass'unt	av'ut	batt'ut	bev'ut

	CADERE	CAMBIARE	CAPIRE	CARPIRE	CEDERE	CHIEDERE
V.IND.PRES.1S	k'ado	k'ambjo	kap'isko	karp'isko	tʃ'edo	kj'edo
V.IND.PRES.2S	k'adi	k'ambi	kap'iʃʃi	karp'iʃʃi	tʃ'edi	kj'edi
V.IND.PRES.3S	k'ade	k'ambja	kap'iʃʃe	karp'iʃʃe	tʃ'ede	kj'ede
V.MC_4 (V.IND.PRES.1P)	kadj	kambj	kapj	karpj	tʃedj	kjedj
V.IND.PRES.2P	kad'e	kambj'a	kap'i	karp'i	tʃed'e	kjed'e
V.IND.PRES.3P	k'ado	k'ambja	kap'isko	karp'isko	tʃ'edo	kj'edo
V.MC_7 (V.IND.IMPF.1S)	kad'ev	kambj'av	kap'iv	karp'iv	tʃed'ev	kjed'ev
V.MC_8 (V.IND.IMPF.1P)	kade	kambja	kapi	karpi	tʃede	kjede
V.IND.PREM.1S	k'addi	kambj'aj	kap'ij	karp'ij	tʃed'etti tʃed'ej	kj'ezi
V.MC_10 (V.IND.PREM.2S)	kad'e	kambj'a	kap'i	karp'i	tʃed'e	kjed'e
V.IND.PREM.3S	k'adde	kambj'ɔ	kap'i	karp'i	tʃed'e tʃed'ette	kj'eze
V.IND.PREM.1P	kad'e	kambj'a	kap'i	karp'i	tʃed'e	kjed'e
V.IND.PREM.3P	k'adder	kambj'aron	kap'iron	karp'iron	tʃed'etter tʃed'eron	kj'ezer
V.MC_14 (V.IND.FUT.1S)	kad	kambje	kapi	karpi	tʃede	kjede
V.MC_15 (V.CONG.PRES.1S)	k'ada	k'ambi	kap'iska	karp'iska	tʃ'eda	kj'eda
V.IMP.2S	k'adi	k'ambja	kap'iʃʃi	karp'iʃʃi	tʃ'edi	kj'edi
V.IMP.2P	kad'e	kambj'a	kap'i	karp'i	tʃed'e	kjed'e
V.GERU	kad'e	kambj'a	kap'e	karp'e	tʃed'e	kjed'e
V.INF	kad'e	kambj'a	kap'i	karp'i	tʃ'ede	kj'ede
V.MC_20 (V.PART.PRES.S)	kad'e	kambj'a	kap'e	karp'e	tʃed'e	kjed'e
V.MC_21 (V.PART.PASS.M.S)	kad'ut	kambj'at	kap'it	karp'it	tʃed'ut	kj'ɛst

	CHIUDERE	CINGERE	COGLIERE	COMPIERE	CONCEDERE	CONCERNERE
V.IND.PRES.1S	kj'udo	tʃ'ingo	k'ɔlgo	k'ompjo	kontʃ'edo	kontʃ'ɛrno
V.IND.PRES.2S	kj'udi	tʃ'indʒi	k'ɔʎʎi	k'ompi	kontʃ'edi	kontʃ'ɛrni
V.IND.PRES.3S	kj'ude	tʃ'indʒe	k'ɔʎʎe	k'ompje	kontʃ'ede	kontʃ'ɛrne
V.MC_4 (V.IND.PRES.1P)	kjudj	tʃindʒ	koʎʎ	kompj	kontʃedj	kontʃernj
V.IND.PRES.2P	kjud'e	tʃindʒ'e	koʎʎ'e	komp'i	kontʃed'e	kontʃern'e
V.IND.PRES.3P	kj'udo	tʃ'ingo	k'ɔlgo	k'ompjo	kontʃ'edo	kontʃ'ɛrno
V.MC_7 (V.IND.IMPF.1S)	kjud'ev	tʃindʒ'ev	koʎʎ'ev	komp'iv	kontʃed'ev	kontʃern'ev
V.MC_8 (V.IND.IMPF.1P)	kjude	tʃindʒe	koʎʎe	kompi	kontʃede	kontʃerne
V.IND.PREM.1S	kj'uzi	tʃ'insi	k'ɔlsi	komp'ij	kontʃ'ɛssi	kontʃern'ej
V.MC_10 (V.IND.PREM.2S)	kjud'e	tʃindʒ'e	koʎʎ'e	komp'i	kontʃed'e	kontʃern'e
V.IND.PREM.3S	kj'uze	tʃ'inse	k'ɔlse	komp'i	kontʃ'ɛsse	kontʃern'e
V.IND.PREM.1P	kjud'e	tʃindʒ'e	koʎʎ'e	komp'i	kontʃed'e	kontʃern'e
V.IND.PREM.3P	kj'uzer	tʃ'inser	k'ɔlser	komp'iron	kontʃ'ɛsser	kontʃern'eron
V.MC_14 (V.IND.FUT.1S)	kjude	tʃindʒe	koʎʎe	kompi	kontʃede	kontʃerne
V.MC_15 (V.CONG.PRES.1S)	kj'uda	tʃ'inga	k'ɔlga	k'ompja	kontʃ'eda	kontʃ'ɛrna
V.IMP.2S	kj'udi	tʃ'indʒi	k'ɔʎʎi	k'ompi	kontʃ'edi	kontʃ'ɛrni
V.IMP.2P	kjud'e	tʃindʒ'e	koʎʎ'e	komp'i	kontʃed'e	kontʃern'e
V.GERU	kjud'e	tʃindʒ'e	koʎʎ'e	kompj'e	kontʃed'e	kontʃern'e
V.INF	kj'ude	tʃ'indʒe	k'ɔʎʎe	k'ompje	kontʃ'ede	kontʃ'ɛrne
V.MC_20 (V.PART.PRES.S)	kjud'e	tʃindʒ'e	koʎʎ'e	kompj'e	kontʃed'e	kontʃern'e
V.MC_21 (V.PART.PASS.M.S)	kj'uz	tʃ'int	k'ɔlt	kompj'ut	kontʃ'ɛss	-

	CONDURRE	CONNETTERE	CONOSCERE	CONSUMARE	CONTROLLARE	CONVERTIRE
V.IND.PRES.1S	kond'uko	konn'etto	kon'osko	kons'umo	kontr'ɔllo	konv'ɛrto
V.IND.PRES.2S	kond'utʃi	konn'etti	kon'oʃʃi	kons'umi	kontr'ɔlli	konv'ɛrti
V.IND.PRES.3S	kond'utʃe	konn'ette	kon'oʃʃe	kons'uma	kontr'ɔlla	konv'ɛrte
V.MC_4 (V.IND.PRES.1P)	kondutʃ	konnettj	konoʃʃ	konsumj	kontrollj	konvertj
V.IND.PRES.2P	kondutʃ'e	konnett'e	konoʃʃ'e	konsum'a	kontroll'a	konvert'i
V.IND.PRES.3P	kond'uko	konn'etto	kon'osko	kons'uma	kontr'ɔlla	konv'ɛrto
V.MC_7 (V.IND.IMPF.1S)	kondutʃ'ev	konnett'ev	konoʃʃ'ev	konsum'av	kontroll'av	konvert'iv
V.MC_8 (V.IND.IMPF.1P)	kondutʃe	konnette	konoʃʃe	konsuma	kontrolla	konverti
V.IND.PREM.1S	kond'ussi	konnett'ej	kon'obbi	konsum'aj	kontroll'aj	konvert'ij
V.MC_10 (V.IND.PREM.2S)	kondutʃ'e	konnett'e	konoʃʃ'e	konsum'a	kontroll'a	konvert'i
V.IND.PREM.3S	kond'usse	konnett'e	kon'obbe	konsum'ɔ	kontroll'ɔ	konvert'i
V.IND.PREM.1P	kondutʃ'e	konnett'e	konoʃʃ'e	konsum'a	kontroll'a	konvert'i
V.IND.PREM.3P	kond'usser	konnett'eron	kon'obber	konsum'aron	kontroll'aron	konvert'iron
V.MC_14 (V.IND.FUT.1S)	kondur	konnette	konoʃʃe	konsume	kontrolle	konverti
V.MC_15 (V.CONG.PRES.1S)	kond'uka	konn'etta	kon'oska	kons'umi	kontr'ɔlli	konv'ɛrta
V.IMP.2S	kond'utʃi	konn'etti	kon'oʃʃi	kons'uma	kontr'ɔlla	konv'ɛrti
V.IMP.2P	kondutʃ'e	konnett'e	konoʃʃ'e	konsum'a	kontroll'a	konvert'i
V.GERU	kondutʃ'e	konnett'e	konoʃʃ'e	konsum'a	kontroll'a	konvert'e
V.INF	kond'ur	konn'ette	kon'oʃʃe	konsum'a	kontroll'a	konvert'i
V.MC_20 (V.PART.PRES.S)	kondutʃ'e	konnett'e	konoʃʃ'e	konsum'a	kontroll'a	konvert'e
V.MC_21 (V.PART.PASS.M.S)	kond'ott	konn'ess	konoʃʃ'ut	konsum'at	kontroll'at	konvert'it

	COPIARE	CORRERE	COSTRUIRE	CREDERE	CREPARE	CRESCERE
V.IND.PRES.1S	k'ɔpjo	k'orro	kostru'isko	kr'edo	kr'epo	kr'esko
V.IND.PRES.2S	k'ɔpi	k'orri	kostru'iʃʃi	kr'edi	kr'epi	kr'eʃʃi
V.IND.PRES.3S	k'ɔpja	k'orre	kostru'iʃʃe	kr'ede	kr'epa	kr'eʃʃe
V.MC_4 (V.IND.PRES.1P)	kopj	korrj	kostruj	kredj	krepj	kreʃʃ
V.IND.PRES.2P	kopj'a	korr'e	kostru'i	kred'e	krep'a	kreʃʃ'e
V.IND.PRES.3P	k'ɔpja	k'orro	kostru'isko	kr'edo	kr'epa	kr'esko
V.MC_7 (V.IND.IMPF.1S)	kopj'av	korr'ev	kostru'iv	kred'ev	krep'av	kreʃʃ'ev
V.MC_8 (V.IND.IMPF.1P)	kopja	korre	kostrui	krede	krepa	kreʃʃe
V.IND.PREM.1S	kopj'aj	k'orsi	kostru'ij	kred'etti kred'ej	krep'aj	kr'ebbi
V.MC_10 (V.IND.PREM.2S)	kopj'a	korr'e	kostru'i	kred'e	krep'a	kreʃʃ'e
V.IND.PREM.3S	kopj'ɔ	k'orse	kostru'i	kred'ette kred'e	krep'ɔ	kr'ebbe
V.IND.PREM.1P	kopj'a	korr'e	kostru'i	kred'e	krep'a	kreʃʃ'e
V.IND.PREM.3P	kopj'aron	k'orser	kostru'iron	kred'etter kred'eron	krep'aron	kr'ebber
V.MC_14 (V.IND.FUT.1S)	kopje	korre	kostrui	krede	krepe	kreʃʃe
V.MC_15 (V.CONG.PRES.1S)	k'ɔpi	k'orra	kostru'iska	kr'eda	kr'epi	kr'eska
V.IMP.2S	k'ɔpja	k'orri	kostru'iʃʃi	kr'edi	kr'epa	kr'eʃʃi
V.IMP.2P	kopj'a	korr'e	kostru'i	kred'e	krep'a	kreʃʃ'e
V.GERU	kopj'a	korr'e	kostru'e	kred'e	krep'a	kreʃʃ'e
V.INF	kopj'a	k'orre	kostru'i	kr'ede	krep'a	kr'eʃʃe
V.MC_20 (V.PART.PRES.S)	kopj'a	korr'e	kostru'e	kred'e	krep'a	kreʃʃ'e
V.MC_21 (V.PART.PASS.M.S)	kopj'at	k'ors	kostru'it	kred'ut	krep'at	kreʃʃ'ut

	CUOCERE	DARE	DEVOLVERE	DIFENDERE	DIPINGERE	DIRE
V.IND.PRES.1S	kw'ɔtʃo	d'ɔ	dev'ɔlvo	dif'endo	dip'iŋgo	d'iko
V.IND.PRES.2S	kw'ɔtʃi	d'aj	dev'ɔlvi	dif'endi	dip'indʒi	d'itʃi
V.IND.PRES.3S	kw'ɔtʃe	d'a	dev'ɔlve	dif'ende	dip'indʒe	d'itʃe
V.MC_4 (V.IND.PRES.1P)	kotʃ kwotʃ	dj	devolvj	difendj	dipindʒ	ditʃ
V.IND.PRES.2P	kwotʃ'e kotʃ'e	d'a	devolv'e	difend'e	dipindʒ'e	d'i
V.IND.PRES.3P	kw'ɔtʃo	d'an	dev'ɔlvo	dif'endo	dip'iŋgo	d'iko
V.MC_7 (V.IND.IMPF.1S)	kotʃ'ev kwotʃ'ev	d'av	devolv'ev	difend'ev	dipindʒ'ev	ditʃ'ev
V.MC_8 (V.IND.IMPF.1P)	kotʃe kwotʃe	da	devolve	difende	dipindʒe	ditʃe
V.IND.PREM.1S	k'ɔssi	dj'edi	devolv'ej devolv'etti	dif'ezi	dip'insi	d'issi
V.MC_10 (V.IND.PREM.2S)	kwotʃ'e kotʃ'e	d'e	devolv'e	difend'e	dipindʒ'e	ditʃ'e
V.IND.PREM.3S	k'ɔsse	dj'ede	devolv'ette devolv'e	dif'eze	dip'inse	d'isse
V.IND.PREM.1P	kwotʃ'e kotʃ'e	d'e	devolv'e	difend'e	dipindʒ'e	ditʃ'e
V.IND.PREM.3P	k'ɔsser	dj'eder	devolv'eron devolv'etter	dif'ezer	dip'inser	d'isser
V.MC_14 (V.IND.FUT.1S)	kotʃe kwotʃe	da	devolve	difende	dipindʒe	di
V.MC_15 (V.CONG.PRES.1S)	kw'ɔtʃa	d'ia	dev'ɔlva	dif'enda	dip'iŋga	d'ika
V.IMP.2S	kw'ɔtʃi	d'a d'aj	dev'ɔlvi	dif'endi	dip'indʒi	d'i
V.IMP.2P	kwotʃ'e kotʃ'e	d'a	devolv'e	difend'e	dipindʒ'e	d'i
V.GERU	kwotʃ'e kotʃ'e	d'a	devolv'e	difend'e	dipindʒ'e	ditʃ'e
V.INF	kw'ɔtʃe	d'a	dev'ɔlve	dif'ende	dip'indʒe	d'i
V.MC_20 (V.PART.PRES.S)	kwotʃ'e kotʃ'e	d'a	devolv'e	difend'e	dipindʒ'e	ditʃ'e
V.MC_21 (V.PART.PASS.M.S)	k'ott	d'at	devol'ut	dif'ez	dip'int	d'ett

	DIRIGERE	DISCUTERE	DISTINGUERE	DISTRUGGERE	DOLERE	DOMARE
V.IND.PRES.1S	dir'igo	disk'uto	dist'iŋgwo	distr'uggo	d'ɔlgo	d'ɔmo
V.IND.PRES.2S	dir'idʒi	disk'uti	dist'iŋgwi	distr'uddʒi	dw'ɔli	d'ɔmi
V.IND.PRES.3S	dir'idʒe	disk'ute	dist'iŋgwe	distr'uddʒe	dw'ɔle	d'ɔma
V.MC_4 (V.IND.PRES.1P)	diridʒ	diskutj	distiŋgwj	distruddʒ	dolj	domj
V.IND.PRES.2P	diridʒ'e	diskut'e	distiŋgw'e	distruddʒ'e	dol'e	dom'a
V.IND.PRES.3P	dir'igo	disk'uto	dist'iŋgwo	distr'uggo	d'ɔlgo	d'ɔma
V.MC_7 (V.IND.IMPF.1S)	diridʒ'ev	diskut'ev	distiŋgw'ev	distruddʒ'ev	dol'ev	dom'av
V.MC_8 (V.IND.IMPF.1P)	diridʒe	diskute	distiŋgwe	distruddʒe	dole	doma
V.IND.PREM.1S	dir'ɛssi	disk'ussi	dist'insi	distr'ussi	d'olsi	dom'aj
V.MC_10 (V.IND.PREM.2S)	diridʒ'e	diskut'e	distiŋgw'e	distruddʒ'e	dol'e	dom'a
V.IND.PREM.3S	dir'ɛsse	disk'usse	dist'inse	distr'usse	d'olse	dom'ɔ
V.IND.PREM.1P	diridʒ'e	diskut'e	distiŋgw'e	distruddʒ'e	dol'e	dom'a
V.IND.PREM.3P	dir'ɛsser	disk'usser	dist'inser	distr'usser	d'olser	dom'aron
V.MC_14 (V.IND.FUT.1S)	diridʒe	diskute	distiŋgwe	distruddʒe	dor	dome
V.MC_15 (V.CONG.PRES.1S)	dir'iga	disk'uta	dist'iŋgwa	distr'ugga	d'ɔlga	d'ɔmi
V.IMP.2S	dir'idʒi	disk'uti	dist'iŋgwi	distr'uddʒi	dw'ɔli	d'ɔma
V.IMP.2P	diridʒ'e	diskut'e	distiŋgw'e	distruddʒ'e	dol'e	dom'a
V.GERU	diridʒ'e	diskut'e	distiŋgw'e	distruddʒ'e	dol'e	dom'a
V.INF	dir'idʒe	disk'ute	dist'iŋgwe	distr'uddʒe	dol'e	dom'a
V.MC_20 (V.PART.PRES.S)	diridʒ'e	diskut'e	distiŋgw'e	distruddʒ'e	dol'e	dom'a
V.MC_21 (V.PART.PASS.M.S)	dir'ɛtt	disk'uss	dist'int	distr'utt	dol'ut	dom'at

	DOVERE	EMETTERE	ESIGERE	ESISTERE	ESPELLERE	ESPRIMERE
V.IND.PRES.1S	d'ebbo d'evo	em'etto	ez'igo	ez'isto	esp'ɛllo	espr'imo
V.IND.PRES.2S	d'evi	em'etti	ez'idʒi	ez'isti	esp'ɛlli	espr'imi
V.IND.PRES.3S	d'eve	em'ette	ez'idʒe	ez'iste	esp'ɛlle	espr'ime
V.MC_4 (V.IND.PRES.1P)	dobbj	emettj	ezidʒ	ezistj	espellj	esprimj
V.IND.PRES.2P	dov'e	emett'e	ezidʒ'e	ezist'e	espell'e	esprim'e
V.IND.PRES.3P	d'ebbo d'evo	em'etto	ez'igo	ez'isto	esp'ɛllo	espr'imo
V.MC_7 (V.IND.IMPF.1S)	dov'ev	emett'ev	ezidʒ'ev	ezist'ev	espell'ev	esprim'ev
V.MC_8 (V.IND.IMPF.1P)	dove	emette	ezidʒe	eziste	espelle	esprime
V.IND.PREM.1S	dov'etti dov'ej	em'izi	ezidʒ'ej	ezist'ej	esp'ulsi	espr'ɛssi
V.MC_10 (V.IND.PREM.2S)	dov'e	emett'e	ezidʒ'e	ezist'e	espell'e	esprim'e
V.IND.PREM.3S	dov'ette dov'e	em'ize	ezidʒ'e	ezist'e	esp'ulse	espr'ɛsse
V.IND.PREM.1P	dov'e	emett'e	ezidʒ'e	ezist'e	espell'e	esprim'e
V.IND.PREM.3P	dov'eron dov'etter	em'izer	ezidʒ'eron	ezist'eron	esp'ulser	espr'ɛsser
V.MC_14 (V.IND.FUT.1S)	dov	emette	ezidʒe	eziste	espelle	esprime
V.MC_15 (V.CONG.PRES.1S)	d'ebba d'eva	em'etta	ez'iga	ez'ista	esp'ɛlla	espr'ima
V.IMP.2S	-	em'etti	ez'idʒi	ez'isti	esp'ɛlli	espr'imi
V.IMP.2P	-	emett'e	ezidʒ'e	ezist'e	espell'e	esprim'e
V.GERU	dov'e	emett'e	ezidʒ'e	ezist'e	espell'e	esprim'e
V.INF	dov'e	em'ette	ez'idʒe	ez'iste	esp'ɛlle	espr'ime
V.MC_20 (V.PART.PRES.S)	-	emett'e	ezidʒ'e	ezist'e	espell'e	esprim'e
V.MC_21 (V.PART.PASS.M.S)	dov'ut	em'ess	ez'att	ezist'it	esp'uls	espr'ɛss

	ESSERE	ESTINGUERE	FARE	FERIRE	FONDERE	FREGARE
V.IND.PRES.1S	s'ono	est'iŋgwo	f'attʃo	fer'isko	f'ondo	fr'ego
V.IND.PRES.2S	s'ɛj	est'iŋgwi	f'aj	fer'iʃʃi	f'ondi	fr'egi
V.IND.PRES.3S	'ɛ	est'iŋgwe	f'a	fer'iʃʃe	f'onde	fr'ega
V.MC_4 (V.IND.PRES.1P)	sj	estiŋgwj	fattʃ	ferj	fondj	fregj
V.IND.PRES.2P	sj'ɛ	estiŋgw'e	f'a	fer'i	fond'e	freg'a
V.IND.PRES.3P	s'o	est'iŋgwo	f'an	fer'isko	f'ondo	fr'ega
V.MC_7 (V.IND.IMPF.1S)	'ɛr	estiŋgw'ev	fatʃ'ev	fer'iv	fond'ev	freg'av
V.MC_8 (V.IND.IMPF.1P)	era	estiŋgwe	fatʃe	feri	fonde	frega
V.IND.PREM.1S	f'uj	est'insi	f'etʃi	fer'ij	f'uzi	freg'aj
V.MC_10 (V.IND.PREM.2S)	f'o	estiŋgw'e	fatʃ'e	fer'i	fond'e	freg'a
V.IND.PREM.3S	f'u	est'inse	f'etʃe	fer'i	f'uze	freg'ɔ
V.IND.PREM.1P	f'u	estiŋgw'e	fatʃ'e	fer'i	fond'e	freg'a
V.IND.PREM.3P	f'uron	est'inser	f'etʃer	fer'iron	f'uzer	freg'aron
V.MC_14 (V.IND.FUT.1S)	sa	estiŋgwe	fa	feri	fonde	frege
V.MC_15 (V.CONG.PRES.1S)	s'ia	est'iŋgwa	f'attʃa	fer'iska	f'onda	fr'egi
V.IMP.2S	s'ij	est'iŋgwi	f'a f'aj	fer'iʃʃi	f'ondi	fr'ega
V.IMP.2P	sj'a	estiŋgw'e	f'a	fer'i	fond'e	freg'a
V.GERU	ess'ɛ	estiŋgw'e	fatʃ'e	fer'e	fond'e	freg'a
V.INF	'ɛsse	est'iŋgwe	f'a	fer'i	f'onde	freg'a
V.MC_20 (V.PART.PRES.S)	ess'ɛ	estiŋgw'e	fatʃ'e	fer'e	fond'e	freg'a
V.MC_21 (V.PART.PASS.M.S)	st'at	est'int	f'att	fer'it	f'uz	freg'at

	FRIGGERE	FUGGIRE	GEMERE	GODERE	INCUTERE	INFRANGERE
V.IND.PRES.1S	fr'iggo	f'uggo	dʒ'emo	g'ɔdo	ink'uto	iɱfr'aŋgo
V.IND.PRES.2S	fr'iddʒi	f'uddʒi	dʒ'emi	g'ɔdi	ink'uti	iɱfr'andʒi
V.IND.PRES.3S	fr'iddʒe	f'uddʒe	dʒ'eme	g'ɔde	ink'ute	iɱfr'andʒe
V.MC_4 (V.IND.PRES.1P)	friddʒ	fuddʒ	dʒemj	godj	inkutj	iɱfrandʒ
V.IND.PRES.2P	friddʒ'e	fuddʒ'i	dʒem'e	god'e	inkut'e	iɱfrandʒ'e
V.IND.PRES.3P	fr'iggo	f'uggo	dʒ'emo	g'ɔdo	ink'uto	iɱfr'aŋgo
V.MC_7 (V.IND.IMPF.1S)	friddʒ'ev	fuddʒ'iv	dʒem'ev	god'ev	inkut'ev	iɱfrandʒ'ev
V.MC_8 (V.IND.IMPF.1P)	friddʒe	fuddʒi	dʒeme	gode	inkute	iɱfrandʒe
V.IND.PREM.1S	fr'issi	fuddʒ'ij	dʒem'ej dʒem'etti	god'ej	ink'ussi	iɱfr'ansi
V.MC_10 (V.IND.PREM.2S)	friddʒ'e	fuddʒ'i	dʒem'e	god'e	inkut'e	iɱfrandʒ'e
V.IND.PREM.3S	fr'isse	fuddʒ'i	dʒem'e dʒem'ette	god'e	ink'usse	iɱfr'anse
V.IND.PREM.1P	friddʒ'e	fuddʒ'i	dʒem'e	god'e	inkut'e	iɱfrandʒ'e
V.IND.PREM.3P	fr'isser	fuddʒ'iron	dʒem'eron dʒem'etter	god'eron	ink'usser	iɱfr'anser
V.MC_14 (V.IND.FUT.1S)	friddʒe	fuddʒi	dʒeme	god	inkute	iɱfrandʒe
V.MC_15 (V.CONG.PRES.1S)	fr'igga	f'ugga	dʒ'ema	g'ɔda	ink'uta	iɱfr'aŋga
V.IMP.2S	fr'iddʒi	f'uddʒi	dʒ'emi	g'ɔdi	ink'uti	iɱfr'andʒi
V.IMP.2P	friddʒ'e	fuddʒ'i	dʒem'e	god'e	inkut'e	iɱfrandʒ'e
V.GERU	friddʒ'e	fuddʒ'e	dʒem'e	god'e	inkut'e	iɱfrandʒ'e
V.INF	fr'iddʒe	fuddʒ'i	dʒ'eme	god'e	ink'ute	iɱfr'andʒe
V.MC_20 (V.PART.PRES.S)	friddʒ'e	fuddʒ'e	dʒem'e	god'e	inkut'e	iɱfrandʒ'e
V.MC_21 (V.PART.PASS.M.S)	fr'itt	fuddʒ'it	dʒem'ut	god'ut	ink'uss	iɱfr'ant

	LEDERE	LEGGERE	MASTICARE	MERITARE	METTERE	MORDERE
V.IND.PRES.1S	l'edo	l'ɛggo	m'astiko	m'ɛrito	m'etto	m'ɔrdo
V.IND.PRES.2S	l'edi	l'ɛddʒi	m'astiki	m'ɛriti	m'etti	m'ɔrdi
V.IND.PRES.3S	l'ede	l'ɛddʒe	m'astika	m'ɛrita	m'ette	m'ɔrde
V.MC_4 (V.IND.PRES.1P)	ledj	leddʒ	mastikj	meritj	mettj	mordj
V.IND.PRES.2P	led'e	leddʒ'e	mastik'a	merit'a	mett'e	mord'e
V.IND.PRES.3P	l'edo	l'ɛggo	m'astika	m'ɛrita	m'etto	m'ɔrdo
V.MC_7 (V.IND.IMPF.1S)	led'ev	leddʒ'ev	mastik'av	merit'av	mett'ev	mord'ev
V.MC_8 (V.IND.IMPF.1P)	lede	leddʒe	mastika	merita	mette	morde
V.IND.PREM.1S	l'ezi	l'ɛssi	mastik'aj	merit'aj	m'izi	m'ɔrsi
V.MC_10 (V.IND.PREM.2S)	led'e	leddʒ'e	mastik'a	merit'a	mett'e	mord'e
V.IND.PREM.3S	l'eze	l'ɛsse	mastik'ɔ	merit'ɔ	m'ize	m'ɔrse
V.IND.PREM.1P	led'e	leddʒ'e	mastik'a	merit'a	mett'e	mord'e
V.IND.PREM.3P	l'ezer	l'ɛsser	mastik'aron	merit'aron	m'izer	m'ɔrser
V.MC_14 (V.IND.FUT.1S)	lede	leddʒe	mastike	merite	mette	morde
V.MC_15 (V.CONG.PRES.1S)	l'eda	l'ɛgga	m'astiki	m'ɛriti	m'etta	m'ɔrda
V.IMP.2S	l'edi	l'ɛddʒi	m'astika	m'ɛrita	m'etti	m'ɔrdi
V.IMP.2P	led'e	leddʒ'e	mastik'a	merit'a	mett'e	mord'e
V.GERU	led'e	leddʒ'e	mastik'a	merit'a	mett'e	mord'e
V.INF	l'ede	l'ɛddʒe	mastik'a	merit'a	m'ette	m'ɔrde
V.MC_20 (V.PART.PRES.S)	led'e	leddʒ'e	mastik'a	merit'a	mett'e	mord'e
V.MC_21 (V.PART.PASS.M.S)	l'ez	l'ɛtt	mastik'at	merit'at	m'ess	m'ɔrs

	MORIRE	MUOVERE	NASCERE	NASCONDERE	OSARE	PARERE
V.IND.PRES.1S	mw'ɔjo	mw'ɔvo	n'asko	nask'ondo	'ɔzo	p'ajo
V.IND.PRES.2S	mw'ɔri	mw'ɔvi	n'aʃʃi	nask'ondi	'ɔzi	p'ari
V.IND.PRES.3S	mw'ɔre	mw'ɔve	n'aʃʃe	nask'onde	'ɔza	p'are
V.MC_4 (V.IND.PRES.1P)	morj	mwovj movj	naʃʃ	naskondj	ozj	paj
V.IND.PRES.2P	mor'i	mwov'e mov'e	naʃʃ'e	naskond'e	oz'a	par'e
V.IND.PRES.3P	mw'ɔjo	mw'ɔvo	n'asko	nask'ondo	'ɔza	p'ajo
V.MC_7 (V.IND.IMPF.1S)	mor'iv	mwov'ev mov'ev	naʃʃ'ev	naskond'ev	oz'av	par'ev
V.MC_8 (V.IND.IMPF.1P)	mori	mwove move	naʃʃe	naskonde	oza	pare
V.IND.PREM.1S	mor'ij	m'ɔssi	n'akkwi	nask'ozi	oz'aj	p'arvi
V.MC_10 (V.IND.PREM.2S)	mor'i	mwov'e mov'e	naʃʃ'e	naskond'e	oz'a	par'e
V.IND.PREM.3S	mor'i	m'ɔsse	n'akkwe	nask'oze	oz'ɔ	p'arve
V.IND.PREM.1P	mor'i	mwov'e mov'e	naʃʃ'e	naskond'e	oz'a	par'e
V.IND.PREM.3P	mor'iron	m'ɔsser	n'akkwer	nask'ozer	oz'aron	p'arver
V.MC_14 (V.IND.FUT.1S)	mori	mwove move	naʃʃe	naskonde	oze	par
V.MC_15 (V.CONG.PRES.1S)	mw'ɔja	mw'ɔva	n'aska	nask'onda	'ɔzi	p'aja
V.IMP.2S	mw'ɔri	mw'ɔvi	n'aʃʃi	nask'ondi	'ɔza	p'ari
V.IMP.2P	mor'i	mwov'e mov'e	naʃʃ'e	naskond'e	oz'a	par'e
V.GERU	mor'e	mwov'e mov'e	naʃʃ'e	naskond'e	oz'a	par'e
V.INF	mor'i	mw'ɔve	n'aʃʃe	nas'konde	oz'a	par'e
V.MC_20 (V.PART.PRES.S)	mor'e	mwov'e mov'e	naʃʃ'e	naskond'e	oz'a	par'e
V.MC_21 (V.PART.PASS.M.S)	m'ɔrt	m'ɔss	n'at	nask'ɔst	oz'at	p'ars

	PASCERE	PERCUOTERE	PERDERE	PIACERE	PORGERE	PORRE
V.IND.PRES.1S	p'asko	perkw'ɔto	p'ɛrdo	pj'attʃo	p'ɔrgo	p'oŋgo
V.IND.PRES.2S	p'aʃʃi	perkw'ɔti	p'ɛrdi	pj'atʃi	p'ɔrdʒi	p'oni
V.IND.PRES.3S	p'aʃʃe	perkw'ɔte	p'ɛrde	pj'atʃe	p'ɔrdʒe	p'one
V.MC_4 (V.IND.PRES.1P)	paʃʃ	perkwotj	perdj	pjatʃ	pordʒ	ponj
V.IND.PRES.2P	paʃʃ'e	perkwot'e	perd'e	pjatʃ'e	pordʒ'e	pon'e
V.IND.PRES.3P	p'asko	perkw'ɔto	p'ɛrdo	pj'attʃo	p'ɔrgo	p'oŋgo
V.MC_7 (V.IND.IMPF.1S)	paʃʃ'ev	perkwot'ev	perd'ev	pjatʃ'ev	pordʒ'ev	pon'ev
V.MC_8 (V.IND.IMPF.1P)	paʃʃe	perkwote	perde	pjatʃe	pordʒe	pone
V.IND.PREM.1S	paʃʃ'etti paʃʃ'ej	perk'ɔssi	p'ɛrsi	pj'akkwi	p'ɔrsi	p'ozi
V.MC_10 (V.IND.PREM.2S)	paʃʃ'e	perkwot'e	perd'e	pjatʃ'e	pordʒ'e	pon'e
V.IND.PREM.3S	paʃʃ'e paʃʃ'ette	perk'ɔsse	p'ɛrse	pj'akkwe	p'ɔrse	p'oze
V.IND.PREM.1P	paʃʃ'e	perkwot'e	perd'e	pjatʃ'e	pordʒ'e	pon'e
V.IND.PREM.3P	paʃʃ'etter paʃʃ'eron	perk'ɔsser	p'ɛrser	pj'akkwer	p'ɔrser	p'ozer
V.MC_14 (V.IND.FUT.1S)	paʃʃe	perkwote	perde	pjatʃe	pordʒe	por
V.MC_15 (V.CONG.PRES.1S)	p'aska	perkw'ɔta	p'ɛrda	pj'attʃa	p'ɔrga	p'oŋga
V.IMP.2S	p'aʃʃi	perkw'ɔti	p'ɛrdi	pj'atʃi	p'ɔrdʒi	p'oni
V.IMP.2P	paʃʃ'e	perkwot'e	perd'e	pjatʃ'e	pordʒ'e	pon'e
V.GERU	paʃʃ'e	perkwot'e	perd'e	pjatʃ'e	pordʒ'e	pon'e
V.INF	p'aʃʃe	perkw'ɔte	p'ɛrde	pjatʃ'e	p'ɔrdʒe	p'or
V.MC_20 (V.PART.PRES.S)	paʃʃ'e	perkwot'e	perd'e	pjatʃ'e	pordʒ'e	pon'e
V.MC_21 (V.PART.PASS.M.S)	paʃʃ'ut	perk'ɔss	perd'ut	pjatʃ'ut	p'ɔrt	p'ɔst

	POSSEDERE	PREDILIGERE	PRENDERE	PROVVEDERE	PUNGERE	REDIGERE
V.IND.PRES.1S	poss'ɛggo possj'ɛdo	predil'igo	pr'ɛndo	provv'edo	p'uŋgo	red'igo
V.IND.PRES.2S	possj'ɛdi	predil'idʒi	pr'ɛndi	provv'edi	p'undʒi	red'idʒi
V.IND.PRES.3S	possj'ɛde	predil'idʒe	pr'ɛnde	provv'ede	p'undʒe	red'idʒe
V.MC_4 (V.IND.PRES.1P)	possedj	predilidʒ	prendj	provvedj	pundʒ	redidʒ
V.IND.PRES.2P	possed'e	predilidʒ'e	prend'e	provved'e	pundʒ'e	redidʒ'e
V.IND.PRES.3P	poss'ɛggo possj'ɛdo	predil'igo	pr'ɛndo	provv'edo	p'uŋgo	red'igo
V.MC_7 (V.IND.IMPF.1S)	possed'ev	predilidʒ'ev	prend'ev	provved'ev	pundʒ'ev	redidʒ'ev
V.MC_8 (V.IND.IMPF.1P)	possede	predilidʒe	prende	provvede	pundʒe	redidʒe
V.IND.PREM.1S	possed'etti possed'ej	predil'ɛssi	pr'ezi	provv'idi	p'unsi	red'assi
V.MC_10 (V.IND.PREM.2S)	possed'e	predilidʒ'e	prend'e	provved'e	pundʒ'e	redidʒ'e
V.IND.PREM.3S	possed'ette possed'e	predil'ɛsse	pr'eze	provv'ide	p'unse	red'asse
V.IND.PREM.1P	possed'e	predilidʒ'e	prend'e	provved'e	pundʒ'e	redidʒ'e
V.IND.PREM.3P	possed'eron possed'etter	predil'ɛsser	pr'ezer	provv'ider	p'unser	red'asser
V.MC_14 (V.IND.FUT.1S)	possede	predilidʒe	prende	provvede	pundʒe	redidʒe
V.MC_15 (V.CONG.PRES.1S)	poss'ɛgga possj'ɛda	predil'iga	pr'ɛnda	provv'eda	p'uŋga	red'iga
V.IMP.2S	possj'ɛdi	predil'idʒi	pr'ɛndi	provv'edi	p'undʒi	red'idʒi
V.IMP.2P	possed'e	predilidʒ'e	prend'e	provved'e	pundʒ'e	redidʒ'e
V.GERU	possed'e	predilidʒ'e	prend'e	provved'e	pundʒ'e	redidʒ'e
V.INF	possed'e	predil'idʒe	pr'ɛnde	provved'e	p'undʒe	red'idʒe
V.MC_20 (V.PART.PRES.S)	possed'e	predilidʒ'e	prend'e	provved'e	pundʒ'e	redidʒ'e
V.MC_21 (V.PART.PASS.M.S)	possed'ut	predil'ɛtt	pr'ez	provved'ut	p'unt	red'att

	RESTRINGERE	RICEVERE	RIFLETTERE	RIMANERE	RISCUOTERE	RISOLVERE
V.IND.PRES.1S	restr'iŋgo	ritʃ'evo	rifl'ɛtto	rim'aŋgo	riskw'ɔto	ris'ɔlvo
V.IND.PRES.2S	restr'indʒi	ritʃ'evi	rifl'ɛtti	rim'ani	riskw'ɔti	ris'ɔlvi
V.IND.PRES.3S	restr'indʒe	ritʃ'eve	rifl'ɛtte	rim'ane	riskw'ɔte	ris'ɔlve
V.MC_4 (V.IND.PRES.1P)	restrindʒ	ritʃevj	riflettj	rimanj	riskwotj	risolvj
V.IND.PRES.2P	restrindʒ'e	ritʃev'e	riflett'e	riman'e	riskwot'e	risolv'e
V.IND.PRES.3P	restr'iŋgo	ritʃ'evo	rifl'ɛtto	rim'aŋgo	riskw'ɔto	ris'ɔlvo
V.MC_7 (V.IND.IMPF.1S)	restrindʒ'ev	ritʃev'ev	riflett'ev	riman'ev	riskwot'ev	risolv'ev
V.MC_8 (V.IND.IMPF.1P)	restrindʒe	ritʃeve	riflette	rimane	riskwote	risolve
V.IND.PREM.1S	restr'insi	ritʃev'ej ritʃev'etti	riflett'ej	rim'azi	risk'ɔssi	ris'ɔlsi
V.MC_10 (V.IND.PREM.2S)	restrindʒ'e	ritʃev'e	riflett'e	riman'e	riskwot'e	risolv'e
V.IND.PREM.3S	restr'inse	ritʃev'e ritʃev'ette	riflett'e	rim'aze	risk'ɔsse	ris'ɔlse
V.IND.PREM.1P	restrindʒ'e	ritʃev'e	riflett'e	riman'e	riskwot'e	risolv'e
V.IND.PREM.3P	restr'inser	ritʃev'eron ritʃev'etter	riflett'eron	rim'azer	risk'ɔsser	ris'ɔlser
V.MC_14 (V.IND.FUT.1S)	restrindʒe	ritʃeve	riflette	rimar	riskwote	risolve
V.MC_15 (V.CONG.PRES.1S)	restr'iŋga	ritʃ'eva	rifl'ɛtta	rim'aŋga	riskw'ɔta	ris'ɔlva
V.IMP.2S	restr'indʒi	ritʃ'evi	rifl'ɛtti	rim'ani	riskw'ɔti	ris'ɔlvi
V.IMP.2P	restrindʒ'e	ritʃev'e	riflett'e	riman'e	riskwot'e	risolv'e
V.GERU	restrindʒ'e	ritʃev'e	riflett'e	riman'e	riskwot'e	risolv'e
V.INF	restr'indʒe	ritʃ'eve	riflett'e	riman'e	riskw'ɔte	ris'ɔlve
V.MC_20 (V.PART.PRES.S)	restrindʒ'e	ritʃev'e	riflett'e	riman'e	riskwot'e	risolv'e
V.MC_21 (V.PART.PASS.M.S)	ristr'ett	ritʃev'ut	riflett'ut	rim'ast	risk'ɔss	ris'ɔlt

	RISPONDERE	RODERE	ROMPERE	SALIRE	SAPERE	SCEGLIERE
V.IND.PRES.1S	risp'ondo	r'ɔdo	r'ompo	s'algo	s'ɔ	ʃ'elgo
V.IND.PRES.2S	risp'ondi	r'ɔdi	r'ompi	s'ali	s'aj	ʃ'eʎʎi
V.IND.PRES.3S	risp'onde	r'ɔde	r'ompe	s'ale	s'a	ʃ'eʎʎe
V.MC_4 (V.IND.PRES.1P)	rispondj	rodj	rompj	salj	sappj	ʃeʎʎ
V.IND.PRES.2P	rispond'e	rod'e	romp'e	sal'i	sap'e	ʃeʎʎ'e
V.IND.PRES.3P	risp'ondo	r'ɔdo	r'ompo	s'algo	s'an	ʃ'elgo
V.MC_7 (V.IND.IMPF.1S)	rispond'ev	rod'ev	romp'ev	sal'iv	sap'ev	ʃeʎʎ'ev
V.MC_8 (V.IND.IMPF.1P)	risponde	rode	rompe	sali	sape	ʃeʎʎe
V.IND.PREM.1S	risp'ozi	r'ozi	r'uppi	sal'ij	s'eppi	ʃ'elsi
V.MC_10 (V.IND.PREM.2S)	rispond'e	rod'e	romp'e	sal'i	sap'e	ʃeʎʎ'e
V.IND.PREM.3S	risp'oze	r'oze	r'uppe	sal'i	s'eppe	ʃ'else
V.IND.PREM.1P	rispond'e	rod'e	romp'e	sal'i	sap'e	ʃeʎʎ'e
V.IND.PREM.3P	risp'ozer	r'ozer	r'upper	sal'iron	s'epper	ʃ'elser
V.MC_14 (V.IND.FUT.1S)	risponde	rode	rompe	sali	sap	ʃeʎʎe
V.MC_15 (V.CONG.PRES.1S)	risp'onda	r'ɔda	r'ompa	s'alga	s'appja	ʃ'elga
V.IMP.2S	risp'ondi	r'ɔdi	r'ompi	s'ali	s'appi	ʃ'eʎʎi
V.IMP.2P	rispond'e	rod'e	romp'e	sal'i	sappj'a	ʃeʎʎ'e
V.GERU	rispond'e	rod'e	romp'e	sal'e	sap'e	ʃeʎʎ'e
V.INF	risp'onde	r'ɔde	r'ompe	sal'i	sap'e	ʃ'eʎʎe
V.MC_20 (V.PART.PRES.S)	rispond'e	rod'e	romp'e	sal'e	sapj'e	ʃeʎʎ'e
V.MC_21 (V.PART.PASS.M.S)	risp'ɔst	r'ɔz	r'ott	sal'it	sap'ut	ʃ'elt

	SCENDERE	SCRIVERE	SCUOTERE	SEDERE	SENTIRE	SOLERE
V.IND.PRES.1S	ʃ'endo	skr'ivo	skw'ɔto	sj'ɛdo s'ɛggo	s'ento	s'ɔʎʎo
V.IND.PRES.2S	ʃ'endi	skr'ivi	skw'ɔti	sj'ɛdi	s'enti	sw'ɔli
V.IND.PRES.3S	ʃ'ende	skr'ive	skw'ɔte	sj'ɛde	s'ente	sw'ɔle
V.MC_4 (V.IND.PRES.1P)	ʃendj	skrivj	skwotj	sedj	sentj	soʎʎ
V.IND.PRES.2P	ʃend'e	skriv'e	skwot'e	sed'e	sent'i	sol'e
V.IND.PRES.3P	ʃ'endo	skr'ivo	skw'ɔto	sj'ɛdo s'ɛggo	s'ento	s'ɔʎʎo
V.MC_7 (V.IND.IMPF.1S)	ʃend'ev	skriv'ev	skwot'ev	sed'ev	sent'iv	sol'ev
V.MC_8 (V.IND.IMPF.1P)	ʃende	skrive	skwote	sede	senti	sole
V.IND.PREM.1S	ʃ'ezi	skr'issi	sk'ɔssi	sed'etti sed'ej	sent'ij	sol'ej
V.MC_10 (V.IND.PREM.2S)	ʃend'e	skriv'e	skwot'e	sed'e	sent'i	sol'e
V.IND.PREM.3S	ʃ'eze	skr'isse	sk'ɔsse	sed'e sed'ette	sent'i	sol'e
V.IND.PREM.1P	ʃend'e	skriv'e	skwot'e	sed'e	sent'i	sol'e
V.IND.PREM.3P	ʃ'ezer	skr'isser	sk'ɔsser	sed'eron sed'etter	sent'iron	sol'eron
V.MC_14 (V.IND.FUT.1S)	ʃende	skrive	skwote	sede	senti	-
V.MC_15 (V.CONG.PRES.1S)	ʃ'enda	skr'iva	skw'ɔta	s'ɛgga sj'ɛda	s'enta	s'ɔʎʎa
V.IMP.2S	ʃ'endi	skr'ivi	skw'ɔti	sj'ɛdi	s'enti	sw'ɔli
V.IMP.2P	ʃend'e	skriv'e	skwot'e	sed'e	sent'i	sol'e
V.GERU	ʃend'e	skriv'e	skwot'e	sed'e	sent'e	sol'e
V.INF	ʃ'ende	skr'ive	skw'ɔte	sed'e	sent'i	sol'e
V.MC_20 (V.PART.PRES.S)	ʃend'e	skriv'e	skwot'e	sed'e	sent'e	sol'e
V.MC_21 (V.PART.PASS.M.S)	ʃ'ez	skr'itt	sk'ɔss	sed'ut	sent'it	s'ɔlit

	SPARGERE	SPEGNERE	SPLENDERE	STARE	STRIDERE	STRINGERE
V.IND.PRES.1S	sp'argo	sp'ɛŋgo	spl'endo	st'ɔ	str'ido	str'iŋgo
V.IND.PRES.2S	sp'ardʒi	sp'ɛɲɲi	spl'endi	st'aj	str'idi	str'indʒi
V.IND.PRES.3S	sp'ardʒe	sp'ɛɲɲe	spl'ende	st'a	str'ide	str'indʒe
V.MC_4 (V.IND.PRES.1P)	spardʒ	speɲɲ	splendj	stj	stridj	strindʒ
V.IND.PRES.2P	spardʒ'e	speɲɲ'e	splend'e	st'a	strid'e	strindʒ'e
V.IND.PRES.3P	sp'argo	sp'ɛŋgo	spl'endo	st'an	str'ido	str'iŋgo
V.MC_7 (V.IND.IMPF.1S)	spardʒ'ev	speɲɲ'ev	splend'ev	st'av	strid'ev	strindʒ'ev
V.MC_8 (V.IND.IMPF.1P)	spardʒe	speɲɲe	splende	sta	stride	strindʒe
V.IND.PREM.1S	sp'arsi	sp'ɛnsi	splend'ej splend'etti	stj'edi	strid'ej strid'etti	str'insi
V.MC_10 (V.IND.PREM.2S)	spardʒ'e	speɲɲ'e	splend'e	st'e	strid'e	strindʒ'e
V.IND.PREM.3S	sp'arse	sp'ɛnse	splend'ette splend'e	stj'ede	strid'e strid'ette	str'inse
V.IND.PREM.1P	spardʒ'e	speɲɲ'e	splend'e	st'e	strid'e	strindʒ'e
V.IND.PREM.3P	sp'arser	sp'ɛnser	splend'etter splend'eron	stj'eder	strid'etter strid'eron	str'inser
V.MC_14 (V.IND.FUT.1S)	spardʒe	speɲɲe	splende	sta	stride	strindʒe
V.MC_15 (V.CONG.PRES.1S)	sp'arga	sp'ɛŋga	spl'enda	st'ia	str'ida	str'iŋga
V.IMP.2S	sp'ardʒi	sp'ɛɲɲi	spl'endi	st'a st'aj	str'idi	str'indʒi
V.IMP.2P	spardʒ'e	speɲɲ'e	splend'e	st'a	strid'e	strindʒ'e
V.GERU	spardʒ'e	speɲɲ'e	splend'e	st'a	strid'e	strindʒ'e
V.INF	sp'ardʒe	sp'eɲɲe	spl'ende	st'a	str'ide	str'indʒe
V.MC_20 (V.PART.PRES.S)	spardʒ'e	speɲɲ'e	splend'e	st'a	strid'e	strindʒ'e
V.MC_21 (V.PART.PASS.M.S)	sp'ars	sp'ɛnt	-	st'at	-	str'ett

	TACERE	TEMERE	TENDERE	TENERE	TOGLIERE	TORCERE
V.IND.PRES.1S	t'attʃo	t'emo	t'endo	t'eŋgo	t'ɔlgo	t'ɔrko
V.IND.PRES.2S	t'atʃi	t'emi	t'endi	t'jeni	t'ɔʎʎi	t'ɔrtʃi
V.IND.PRES.3S	t'atʃe	t'eme	t'ende	t'jene	t'ɔʎʎe	t'ɔrtʃe
V.MC_4 (V.IND.PRES.1P)	tatʃ	temj	tendj	tenj	toʎʎ	tortʃ
V.IND.PRES.2P	tatʃ'e	tem'e	tend'e	ten'e	toʎʎ'e	tortʃ'e
V.IND.PRES.3P	t'attʃo	t'emo	t'endo	t'eŋgo	t'ɔlgo	t'ɔrko
V.MC_7 (V.IND.IMPF.1S)	tatʃ'ev	tem'ev	tend'ev	ten'ev	toʎʎ'ev	tortʃ'ev
V.MC_8 (V.IND.IMPF.1P)	tatʃe	teme	tende	tene	toʎʎe	tortʃe
V.IND.PREM.1S	t'akkwi	tem'ej tem'etti	t'ezi	t'enni	t'ɔlsi	t'ɔrsi
V.MC_10 (V.IND.PREM.2S)	tatʃ'e	tem'e	tend'e	ten'e	toʎʎ'e	tortʃ'e
V.IND.PREM.3S	t'akkwe	tem'ette tem'e	t'eze	t'enne	t'ɔlse	t'ɔrse
V.IND.PREM.1P	tatʃ'e	tem'e	tend'e	ten'e	toʎʎ'e	tortʃ'e
V.IND.PREM.3P	t'akkwer	tem'etter tem'eron	t'ezer	t'enner	t'ɔlser	t'ɔrser
V.MC_14 (V.IND.FUT.1S)	tatʃe	teme	tende	ter	toʎʎe	tortʃe
V.MC_15 (V.CONG.PRES.1S)	t'attʃa	t'ema	t'enda	t'eŋga	t'ɔlga	t'ɔrka
V.IMP.2S	t'atʃi	t'emi	t'endi	t'jeni	t'ɔʎʎi	t'ɔrtʃi
V.IMP.2P	tatʃ'e	tem'e	tend'e	ten'e	toʎʎ'e	tortʃ'e
V.GERU	tatʃ'e	tem'e	tend'e	ten'e	toʎʎ'e	tortʃ'e
V.INF	tatʃ'e	tem'e	t'ende	ten'e	t'ɔʎʎe	t'ɔrtʃe
V.MC_20 (V.PART.PRES.S)	tatʃ'e	tem'e	tend'e	ten'e	toʎʎ'e	tortʃ'e
V.MC_21 (V.PART.PASS.M.S)	tatʃ'ut	tem'ut	t'ez	ten'ut	t'ɔlt	t'ɔrt

	TRARRE	UDIRE	USCIRE	VALERE	VEDERE	VENIRE
V.IND.PRES.1S	tr'aggo	'ɔdo	'esko	v'algo	v'edo	v'eŋgo
V.IND.PRES.2S	tr'ai	'ɔdi	'eʃʃi	v'ali	v'edi	vj'eni
V.IND.PRES.3S	tr'ae	'ɔde	'eʃʃe	v'ale	v'ede	vj'ene
V.MC_4 (V.IND.PRES.1P)	traj	udj	uʃʃ	valj	vedj	venj
V.IND.PRES.2P	tra'e	ud'i	uʃʃ'i	val'e	ved'e	ven'i
V.IND.PRES.3P	tr'aggo	'ɔdo	'esko	v'algo	v'edo	v'eŋgo
V.MC_7 (V.IND.IMPF.1S)	tra'ev	ud'iv	uʃʃ'iv	val'ev	ved'ev	ven'iv
V.MC_8 (V.IND.IMPF.1P)	trae	udi	uʃʃi	vale	vede	veni
V.IND.PREM.1S	tr'assi	ud'ij	uʃʃ'ij	v'alsi	v'idi	v'enni
V.MC_10 (V.IND.PREM.2S)	tra'e	ud'i	uʃʃ'i	val'e	ved'e	ven'i
V.IND.PREM.3S	tr'asse	ud'i	uʃʃ'i	v'alse	v'ide	v'enne
V.IND.PREM.1P	tra'e	ud'i	uʃʃ'i	val'e	ved'e	ven'i
V.IND.PREM.3P	tr'asser	ud'iron	uʃʃ'iron	v'alser	v'ider	v'enner
V.MC_14 (V.IND.FUT.1S)	trar	udi	uʃʃi	var	ved	ver
V.MC_15 (V.CONG.PRES.1S)	tr'agga	'ɔda	'eska	v'alga	v'eda	v'eŋga
V.IMP.2S	tr'ai	'ɔdi	'eʃʃi	v'ali	v'edi	vj'eni
V.IMP.2P	tra'e	ud'i	uʃʃ'i	val'e	ved'e	ven'i
V.GERU	tra'e	ud'e	uʃʃ'e	val'e	ved'e	ven'e
V.INF	tr'ar	ud'i	uʃʃ'i	val'e	ved'e	ven'i
V.MC_20 (V.PART.PRES.S)	tra'e	ud'e	uʃʃ'e	val'e	ved'e	ven'e
V.MC_21 (V.PART.PASS.M.S)	tra'tt	ud'it	uʃʃ'it	val's	ved'ut v'ist	ven'ut

	VINCERE	VIVERE	VOLERE	VOLGERE
V.IND.PRES.1S	v'iŋko	v'ivo	v'ɔʎʎo	v'ɔlgo
V.IND.PRES.2S	v'intʃi	v'ivi	vw'ɔi	v'ɔldʒi
V.IND.PRES.3S	v'intʃe	v'ive	vw'ɔle	v'ɔldʒe
V.MC_4 (V.IND.PRES.1P)	vintʃ	vivj	voʎʎ	voldʒ
V.IND.PRES.2P	vintʃ'e	viv'e	vol'e	voldʒ'e
V.IND.PRES.3P	v'iŋko	v'ivo	v'ɔʎʎo	v'ɔlgo
V.MC_7 (V.IND.IMPF.1S)	vintʃ'ev	viv'ev	vol'ev	voldʒ'ev
V.MC_8 (V.IND.IMPF.1P)	vintʃe	vive	vole	voldʒe
V.IND.PREM.1S	v'insi	v'issi	v'ɔlli	v'ɔlsi
V.MC_10 (V.IND.PREM.2S)	vintʃ'e	viv'e	vol'e	voldʒ'e
V.IND.PREM.3S	v'inse	v'isse	v'ɔlle	v'ɔlse
V.IND.PREM.1P	vintʃ'e	viv'e	vol'e	voldʒ'e
V.IND.PREM.3P	v'inser	v'isser	v'ɔller	v'ɔlser
V.MC_14 (V.IND.FUT.1S)	vintʃe	viv	vor	voldʒe
V.MC_15 (V.CONG.PRES.1S)	v'iŋka	v'iva	v'ɔʎʎa	v'ɔlga
V.IMP.2S	v'intʃi	v'ivi	v'ɔʎʎi	v'ɔldʒi
V.IMP.2P	vintʃ'e	viv'e	voʎʎ'a	voldʒ'e
V.GERU	vintʃ'e	viv'e	vol'e	voldʒ'e
V.INF	v'intʃe	v'ive	vol'e	v'ɔldʒe
V.MC_20 (V.PART.PRES.S)	vintʃ'e	viv'e	vol'e	voldʒ'e
V.MC_21 (V.PART.PASS.M.S)	v'int	viss'ut	vol'ut	v'ɔlt

5 Appendix: selected metalinguistic definitions

Some key concepts, recurring throughout my work, that may have a specific meaning in *Word and Paradigm* models of morphology, are listed here.

A **grammatical word** of a lexeme L is an instance of L coupled with a set of morphosyntactic features. This is a more restricted definition than **form** of L, in that grammatical words can have the same surface form but be distinct in the morphosyntactic feature sets they realise, as for example *tried* (TRY, PAST) and *tried* (TRY, PAST PARTICIPLE) (Pirrelli and Battista 2000, 307).

An **inflectional paradigm** of a lexeme L is the set of all grammatical words of L. For every form F of L, at least one set of morphosyntactic features exists, which F realises. A set of morphosyntactic features is a **slot** of the paradigm; and the grammatical words of L are **members** of its paradigm (Pirrelli and Battista 2000, 307).

Realisation rules are functions that relate a paradigm slot with a surface form of a given lexeme.

A **stem-formation rule** allows the phonological form of a stem to be deduced from that of some other stem of the same lexeme (Stump 2001, 199).

A **stem-indexing rule** assigns the index to a stem when that index is not explictly assigned by lexical stipulation or by the stem-formation rule which determined its form (Stump 2001, 199).

Stem space is a collection of indexed radicals associated with a lexeme (Bonami and Boyé 2014, 2).

Conclusions

With this work I have carried an analysis of the Italian verbal system. Following a *Word and Paradigm* point of view, and researches who have studied the inflectional morphology with paradigmatic approach, my goal was to build algorithms and programs to calculate relations between the word forms comprising the whole flexion of a sample of Italian verbs. The set of evaluated verbs covers all models of conjugation, including highly irregular verbs.

The contribution to inflectional morphology articulates on these points:

– The analysis is on the phonetic forms, as opposed to orthographic forms. I have thus developed a database for generating forms for all paradigm cells in their phonetic transcription.

– The analysis is fully automated. I have developed all the algorithms needed in Java language, so that after a change in the database (for further lexemes, or possibly correction of mistakes), or even the switch to another set of data, for analysing other languages, the whole computation takes few minutes to run.

– The analysis does not depend on the supposition that inflection happens at the right end of the word, or by suffixation: the algorithms developed can work with discontinuous flexion (as found in Semitic languages, or partially in German and Greek, for example) with the same principles.

Using the software framework I developed, it was possible to extract the following data from the input set:

- Invariance for each paradigm cell: the phonetic material that is characteristic of each cell, independent from the lexeme;
- Grouping of paradigm cells for which every lexeme has the same form minus the invariant segments, "stem" in a broad sense;
- The list of all relations (transformations) between pairs of paradigm cells, based on atomic operations (copy, substitution, deletion, insertion);
- The list of all lexemes adhering to each relation;
- Each relation's entropy value, directly related to form predictability.

In future, many further researches can be carried on with the same procedure and using the same software.

Including every lexeme in the database would permit to associate each form with a frequency value (from a corpus), to fine-tune the entropy calculation on token frequencies: for instance, in the present work my focus was on lexemes which show allomorphy, but obviously relations occurring between Italian verb forms from the conjugation in *-are* are much more predictable than the others.

Of course, analysing the inflection of other languages, including dialects, would be rather simple once added the relevant lexemes to the database. The procedure is flexible enough to work for different models of inflection: not

only suffixal, but also prefixal and discontinuous. The algorithm for comparing forms does not currently handle language-specific transformations (umlauting, diphthongisation, gemination) nor reduplication as basic operations (these transformations emerge from the whole relations), but it can be extended to do so.

Including data sets from different diachronic stages of a language can show how the computed relations between forms and the entropy values of these relations change over time. The attested forms in medieval Italian show more variation (higher entropy values): using the procedure here described, a researcher can list the relations that lowered their entropy, augmenting the predictability of forms and the stability of the paradigm partitioning.

All the computed data can be retrieved from the following URL: http://www.pascoli.it/thesis/thesis_files.zip

References

Ackerman, Farrell, James P. Blevins, and Robert Malouf. 2009. "Parts and Wholes: Implicative Patterns in Inflectional Paradigms." In *Analogy in Grammar: Form and Acquisition*, edited by James P. Blevins and Juliette Blevins, 54–82. Oxford: Oxford University Press.

Albright, Adam C. 2002. *The Identification of Bases in Morphological Paradigms*, PhD Thesis, University of California, Los Angeles.

Anderson, Stephen R. 1992. *A-Morphous Morphology*. Cambridge University Press.

Aronoff, Mark. 1994. *Morphology by Itself: Stems and Inflectional Classes*. Cambridge, MA-London: The MIT Press.

Blevins, James P. 2003. "Stems and Paradigms." *Language* 79 (4): 737–67.

———. 2006. "Word-Based Morphology." *Journal of Linguistics* 42 (03): 531–73.

Bonami, Olivier, and Gilles Boyé. 2001. "Suppletion and Dependency in Inflectional Morphology." *Proceedings of the 8th International HPSG Conference*, edited by Frank van Eynde, Lars Hellan and Dorothee Beermann, 51-70. Stanford: CSLI Publications.

———. 2003. "Supplétion et classes flexionnelles." *Langages* 37 (152): 102–26.

———. 2007. “French Pronominal Clitics and the Design of Paradigm Function Morphology.” In *Proceedings of the Fifth Mediterranean Morphology Meeting*, edited by Geert Booij et al., 291–322. Università di Bologna. http://www3.lingue.unibo.it/ mmm2/wp-content/uploads/2012/09/MMM5-Proceedings_full.pdf.

———. 2014. “De formes en thèmes.” In *Foisonnements morphologiques. Etudes en hommage à Françoise Kerleroux*, edited by Florence Villoing, Sophie David and Sarah Leroy, 17-45. Nanterre: Presses Universitaires de Paris Ouest.

Bonami, Olivier, and Gregory T. Stump. To appear. “Paradigm Function Morphology.” *The Handbook of Morphology,* edited by Andrew Spencer, 2nd Ed. Oxford: Wiley-Blackwell.

Brown, Dunstan. 1998. “Stem Indexing and Morphonological Selection in the Russian Verb: A Network Morphology Account.” In *Models of Inflection*, edited by Ray Fabri, Albert Ortmann, and Teresa Parodi, 161–221. Tübingen: Niemeyer.

Brown, Dunstan, and Andrew Hippisley. 2012. *Network Morphology: A Defaults-Based Theory of Word Structure*. Cambridge: Cambridge University Press.

Bybee, Joan L. 1988. “Morphology as Lexical Organization.” In *Theoretical Morphology: Approaches in Modern Linguistics*, edited by Michael T. Hammond and Michael P. Noonan, 119–41. San Diego: Academic Press.

Corbett, Greville G., and Norman Fraser. 1993. “Network Morphology: A DATR Account of Russian Nominal Inflection.” *Journal of Linguistics* 29 (1): 113–42.

Fortune, George. 1955. *An Analytical Grammar of Shona*. London: Longmans, Green and co.

Giraudo, Hélène, Fabio Montermini, and Vito Pirrelli. 2008. "Processi cognitivi nell'analisi delle classi verbali delliitaliano: un approccio sperimentale." In *Linguaggio e cervello-semantica/Language and the Brain-Semantics. Atti del XLII convegno della Società di Linguistica Italiana, Pisa, Scuola Normale Superiore*, vol. 2 (cd-rom), edited by Pier Marco Bertinetto, Valentina Bambini, and Irene Ricci. Roma:Bulzoni.

Graffi, Giorgio. 1996. "Alcune riflessioni sugli imperativi italiani." In *Italiano e dialetti nel tempo: saggi di grammatica per Giulio C. Lepschy*, edited by Paola Beninca', Guglielmo Cinque, Tullio De Mauro, and Nigel Vincent, 133–48. Roma: Bulzoni.

Halle, Morris, and Alec Marantz. 1993. "Distributed Morphology and the Pieces of Inflection." In *The View from Building 20*, edited by Ken Hale and Samuel Jay Keyser, 111–76. Cambridge, MA: MIT Press.

Hamming, Richard W. 1950. "Error Detecting and Error Correcting Codes." *Bell System Technical Journal* 29 (2): 147–60.

Hockett, Charles Francis. 1954. "Two Models of Grammatical Description." *Morphology: Critical Concepts in Linguistics,* 1:110-138.

Levenshtein, Vladimir I. 1966. "Binary Codes Capable of Correcting Deletions, Insertions and Reversals." In *Soviet Physics Doklady*, 10:707–10.

Lieber, Rochelle. 1992. *Deconstructing Morphology: Word Formation in Syntactic Theory*. Chicago: University of Chicago Press.

Maiden, Martin. 1992. “Irregularity as a Determinant of Morphological Change.” *Journal of Linguistics* 28 (2): 285–312.

Matthews, Peter H. 1972. *Inflectional Morphology: A Theoretical Study Based on Aspects of Latin Verb Conjugation*. Cambridge, UK: Cambridge University Press.

———. 1974. *Morphology*. Cambridge, UK: Cambridge University Press.

Milin, Petar, Victor Kuperman, Aleksandar Kostić, and R. Harald Baayen. 2009. “Words and Paradigms Bit by Bit: An Information-Theoretic Approach to the Processing of Inflection and Derivation.” In *Analogy in Grammar: Form and Acquisition*, edited by James P. Blevins and Juliette Blevins, 214–52. Oxford: Oxford University Press.

Montermini, Fabio, and Olivier Bonami. 2013. “Stem Spaces and Predictability in Verbal Inflection.” *Lingue e Linguaggio* 12 (2): 171–90.

Montermini, Fabio, and Gilles Boyé. 2012. “Stem Relations and Inflection Class Assignment in Italian.” *Word Structure* 5 (1): 69–87.

Moscoso del Prado Martín, Fermín. 2003. *Paradigmatic Structures in Morphological Processing: Computational and Cross-Linguistic Studies*. PhD Thesis, University of Nijmegen

Pirrelli, Vito. 2000. *Paradigmi in morfologia: un approccio interdisciplinare alla flessione verbale dell’italiano*. Pisa-Roma: Istituti editoriali e poligrafici internazionali.

Pirrelli, Vito, and Marco Battista. 2000. “The Paradigmatic Dimension of Stem Allomorphy in Italian Verb Inflection.” *Italian Journal of Linguistics* 12: 307–79.

Shannon, Claude. 1948. “A Mathematical Theory of Communication.” *The Bell System Technical Journal* 27: 379–423, 623–56.

Steele, Susan. 1995. “Towards a Theory of Morphological Information.” *Language*, 260–309.

Stump, Gregory T. 1991. “A Paradigm-Based Theory of Morphosemantic Mismatches.” *Language*, 675–725.

———. 2001. *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge: Cambridge University Press.

Zwicky, Arnold M. 1985. “How to Describe Inflection.” In *Proceedings of the Eleventh Meeting of the Berkeley Linguistics Society*, edited by Mary Niepokuj, Mary Van Clay, Vassiliki Nikiforidou, and Deborah Feder, 372–86. Berkeley: Berkeley Linguistics Society.