A paradigmatic analysis of the Italian verbal derivation

Fabio Montermini
CLLE-ERSS, CNRS
& Université de Toulouse II Jean Jaurès
fabio.montermini@univ-tlse2.fr

Matteo Pascoli
Università di Bologna
matteo.pascoli2@unibo.it

1. Introduction

According to the classification established by Stump (2001), Word-and-Paradigm theories of inflection can be classified as inferential-realizational. In these models, in fact, morphological exponents are not listed in the lexicon along with the corresponding morphosyntactic properties. Rather, they are inferential in that exponents are introduced by rules relating inflected forms with the corresponding lexemes, and are selected by their morphosyntactic properties; they are realizational in that morphosyntactic properties are not added to the word by an exponent, but select the exponents that realize them.

It has been observed (cf. *inter alios* Maiden 1992; Pirrelli and Battista 2000) that, on verbal stems that display allomorphy, alternations show a surprisingly regular distribution, which is not dictated by either the phonological context or the semantic homogeneity. This regularity reflects the organization of the verbal paradigm, or the set of all inflected forms for each lexeme, into *morphomes*, i.e. purely formal entities independent from morphosyntactic features (Aronoff 1994).

In the last twenty years, there has been much interest in the study of the *paradigmatic distribution of allomorphy*, or the way in which variation between forms (the traditional "irregularity") of a paradigm rests on regular patterns.

Practically, these studies aim at analyzing the paradigmatic structure of inflection, i.e. to decompose the paradigm into zones where forms are realized on possibly distinct basic stems, and to examine the formal relations (at a phonological level) between these basic stems, looking for predictability chains allowing to handle both regular and irregular lexemes.

Moreover, it has been shown that, in Romance languages, the formal alternations that apply to inflection are the same that function in derivation (cf. Bonami *et al.* 2009 on French). In the present work, we examine the formal relations between some verbal derivatives in Italian and the basic stems of the related verbs with the goal of extending the study of paradigmatic distribution to derivation.

In Latin, deverbal derivatives in *-tionem* (event/result), *-torem* (agent/instrument), *-tura* (event/result), *-tivus* (relational adjective), *-torium* (adjective/instrument/place) were built on the supine (basic) stem, the same that was used to form the past participle. Italian, like other Romance languages, has inherited from Latin both the process and the derived forms.

(1)	event/result, N	*frittura* 'frying'/'fried food'	*friggere* 'fry', P.P. *fritto*
	event, N	*chiusura* 'closure'	*chiudere* 'close', P.P. *chiuso*
	place, N	*pensatoio* 'thinking place'	*pensare* 'think', P.P. *pensato*
	instrument, N	*rasoio* 'razor'	*radere* 'shave', P.P. *raso*
	(place, N)/A	*uditorio* 'audience'/'auditory'	*udire* 'hear', P.P. *udito*
	A	*illusorio* 'illusory'	*illudere* 'deceive', P.P. *illuso*
	converted, N	*raccolto* 'harvest'	*raccogliere* 'pick up', P.P. *raccolto*
	converted, N	*riassunto* 'summary'	*riassumere* 'sum up', P.P. *riassunto*

Some derivatives, and in some cases the past participle itself, underwent semantic drift. Some (ancient) past participles are no longer connected to a verb and remain in the language as independent adjectives (cf. *solito* 'usual'). Some past participles of existing verbs underwent the same phenomenon (cf. *viso* 'seen' → 'sight' → 'eyes' → 'face' in modern Italian), while being replaced in their past participle function by analogical forms (*veduto*/*visto* 'seen'). In these cases, derivatives do not display a transparent relation with the past participles of the base verbs. Yet, they maintain formal relations, and these relations allow identifying a basic stem, which is by default identical to the basic stem of the past participle, but can possibly be distinct. This basic stem can be related to other basic stems of the base verb.

2. Paradigmatic relations in Italian

A number of recent contributions on the inflectional morphology of Romance languages, and in particular of Italian, showed that, when a lexeme displays allomorphy, stems can be formally connected in predictable ways (cf. Pirrelli and Battista 2000; Montermini and Boyé 2012). Lexemes are identified by a "stem space", i.e. a collection of stems bearing specification about the paradigm area they fill. The links connecting each stem are the same for all lexemes in the language, and some stems are more closely linked than others (cf. Montermini and Boyé 2012 on Italian for details). Under this view, regularity can be viewed as a gradient phenomenon: a lexeme is regular when all its stems are connected in a predictable way; it is irregular when at least one of its stems is unconnected with the rest of the stem space, and thus must be stored independently. Figures 1 and 2 show a portion of the stem space for a regular (traditional 1st class) verb in Italian, *ammirare* ('admire'). S1 corresponds to the simplest stem in the paradigm (used, among others, to form the present 2PL, the imperfect indicative and subjunctive); S7 corresponds to the stem used to form the past participle, and S9 corresponds to the stem used to form the derivatives we referred to above. As the representation we propose shows, we consider that for *ammirare* the three stems are connected by predictable functions. This means that a speaker of Italian is able to recover any form in the paradigm (including derivatives) from any other.

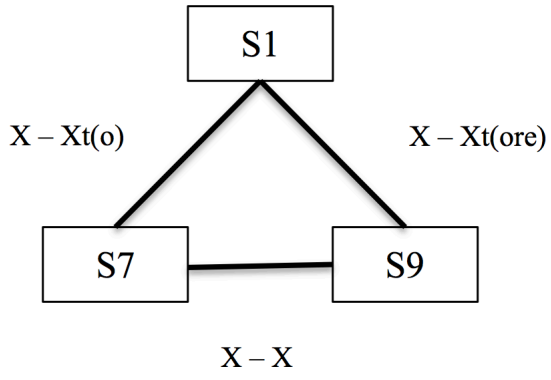

Figure 1: connections between S1, S7 and S9 for a regular verb

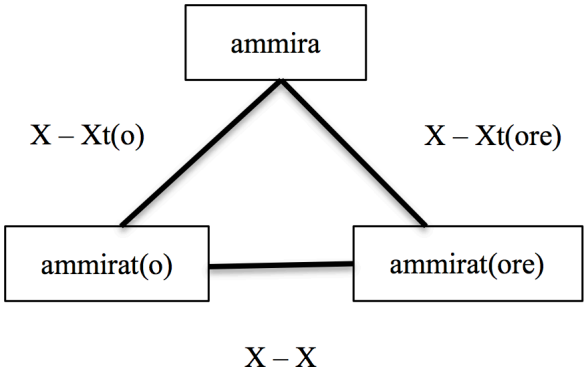

Figure 2: connections between S1, S7 and S9 for a regular verb (*ammirare*)

On the contrary, for an irregular verb, we consider that more than one stem must be stored for a lexeme. It is the case, for instance for *distruggere* (‘destroy’) (Figure 3).

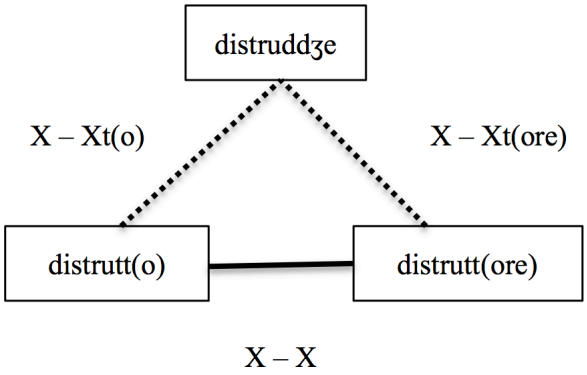

Figure 3: connections between S1, S7 and S9 for an irregular verb (*distruggere*)

As we see, for *distruggere* S7 and S9 are still connected by a predictable function (in this case an identity function), but in any case they are linked to S1. As the examples in (2) show, a S9 ending in [utt] can in fact correspond to S1 having different forms. Thus, we cannot consider that a predictable connection exists between the stems [distruddʒe] and [distrutt].

(2)	S1	S9	Glosses	
	_uddʒe (*distruggete*$_{2.\text{PL.PRES.IND}}$)		_utt (*distruttore*)	‘destroy’/‘destroyer’
	_ui (*costruite*$_{2.\text{PL.PRES.IND}}$)		_utt (*costruttore*)	‘build’/‘constructor’
	_dutʃe (*producete*$_{2.\text{PL.PRES.IND}}$)		_dutt (*produttore*)	‘produce’/‘producer’
	_romp (*interrompete*$_{2.\text{PL.PRES.IND}}$)		_rutt (*interruttore*)	‘interrupt’/‘switch’

Some verbs display a structure which is even more complex. For a verb like *produrre* (‘produce’), for instance, neither of the stems under consideration is connected to the others by predictable functions (Figure 4).

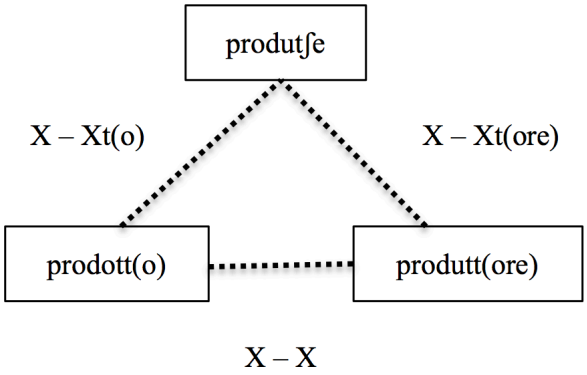

Figure 4: connections between S1, S7 and S9 for an irregular verb (*produrre*)

In synchrony, the stem connections that can be identified for verbs have thus the function of facilitating the recovery of unknown forms by speakers; in diachrony, it can be shown that in some cases paradigms have been restructured following the same lines. Verbs in *_struire* (Figure 5), for instance *costruire* 'build', *istruire* 'educate', etc., display a S9 which is etymological (cf. Latin *construo*, P.P. *constructum*), and thus unconnected to the rest of the paradigm, whereas the S7 has been remodelled in order to be homogeneous with the rest of the paradigm, and is thus connected to the S1.

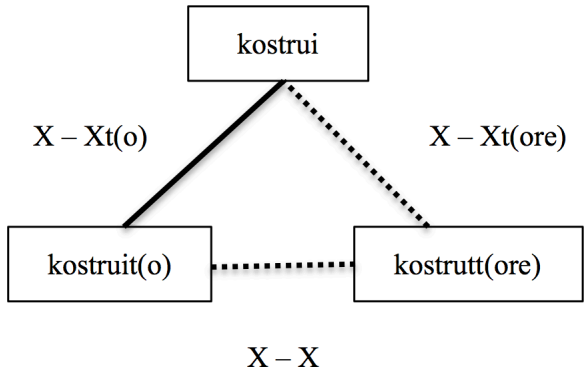

Figure 5: connections between S1, S7 and S9 for a partially regularized verb (*costruire*)

Conversely, for a verb like *scoprire* ('discover') the S7 is etymological (from Lat. *operio*, P.P. *opertum* 'cover' by double prefixation), whereas the S9 has been remodelled on the rest of the paradigm (Figure 6).

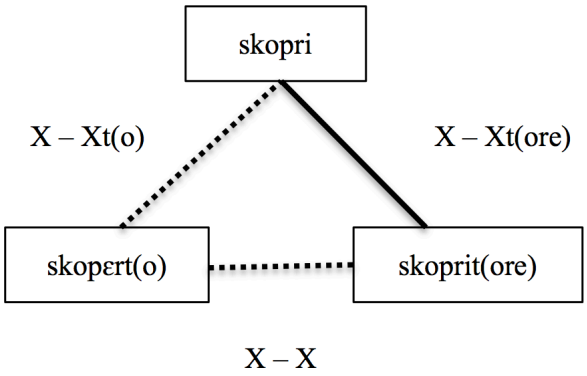

Figure 6: connections between S1, S7 and S9 for a partially regularized verb (*scoprire*)

3. Finding relation between verb stems

We constructed a corpus of 55 forms for 139 Italian verbs. The verb dataset includes all the conjugation models of the GRADIT dictionary (De Mauro 1999). The considered forms include the 46 strictly verbal synthetic forms, 6 participial forms, and 3 deverbal derivatives (*-tivo*, *-tore*, *-zione*), when they exist in the same dictionary for the corresponding verbal lexeme. For the purpose of the present research, the derivatives are considered as part of the verbal paradigm.

We tested a computational technique that could perform two tasks:

(i) for each paradigm cell, separation of the invariant part which is common to all lexemes (we will call this the "model") from the part specific to each lexeme (we will call this the "stem" in a broad sense);
(ii) for each lexeme, collection of the morphophonetic relations between each couple of stems identified in (i).

As an example, the first task should compute the forms *aboliamo, capiamo, cogliamo, dirigiamo, siamo, vediamo* ("we abolish", "we understand", "we pick", "we direct", "we are", "we see"), and give as result: 'X+*iamo*; X=*abol, cap ,cogl, dirig, s, ved*' (the actual computations were performed on phonetically transcribed strings). The second task should take for example two stems for the lexeme DIRIGERE "to direct", [di'rig] and [di'ridʒ] and automatically give the output 'replace [g] with [dʒ] in word final position'.

To perform the tasks described above, we examined the *string distance measure* algorithms. These algorithms aim at quantifying the difference between two given strings, or sequences of symbols.

For instance, one of the distance measures is computed by using the Hamming algorithm (Hamming 1950). This algorithm counts the symbols of the two strings that are different in each given position:

a	l	b	e	r	o
a	l	b	e	r	i
√	√	√	√	√	X
Hamming distance: 1					

Table 1: Hamming distance between *albero* and *alberi*

a	l	b	e	r	o		
v	i	l	l	e	t	t	e
X	X	X	X	X	X	X	X
Hamming distance: 8							

Table 2: Hamming distance between *albero* and *villette*

However, such a simple algorithm is not adequate for our needs. Let us take for instance the strings *abcdefgh* and *bcdefgh*:

a	b	c	d	e	f	g	h
b	c	d	e	f	g	h	
X	X	X	X	X	X	X	X
Hamming distance: 8							

Table 3: Hamming distance between *abcdefgh* and *bcdefgh*

Intuitively, these strings are very similar, much more than the strings *albero* and *villette* in Table 2, for which we get the same result. There exists, however, a different algorithm which can deal with such cases: the Levenshtein algorithm can find similarities and differences in two strings even when they are in different positions (Levenshtein 1966).

The Levenshtein method can analyse the transformation of a string into another by decomposing it into a sequence of basic operations. These basic operations are Insertion (a symbol occurs in the second string but is absent from the first string), Deletion (a symbol occurs in the first string but is absent from the second one), Substitution, (two symbols in the same position are different in the two strings), No-change (the same symbol occurs in the same position in the two strings).

The Levenshtein algorithm is relatively complex and is based on the construction of a rectangular matrix with as many columns as the length of the first string plus one, and as many rows as the length of the string word plus one.

Let us take a simple example to illustrate how the algorithm works for finding the transformation of the string [kaste] into the string [kare]. First, we prepare a matrix, in the way described above, and we fill the first row and the first column with consecutive numbers starting from 0, like in the example below (Figure 7):

•	•	k	a	s	t	e
•	0	1	2	3	4	5
k	1					
a	2					
r	3					
e	4					

Figure 7: setup of the Levenshtein computation

Then, we fill the other cells, from top to bottom and from left to right, with a simple computation: if the symbols labeling the column and the symbol labeling the row coincide, we copy the value found in the cell above at the left of the new cell (Figure 8):

•	•	k	a	s	t	e
•	0	1	2	3	4	5
k	1	**0**				
a	2					
r	3					
e	4					

Figure 8: cell relative to corresponding symbols

If the symbols labeling the column and the row are different, we take the lowest of the values in the above, left, and above left cells (relative to the next empty cell), add 1, and write the resulting value in the new cell (Figure 9).

•	•	k	a	s	t	e
•	0	1	2	3	4	5
k	1	0	**1**			
a	2					
r	3					
e	4					

Figure 9: cell relative to different symbols

Once all the cells in the matrix have been filled, the Levenshtein measure corresponds to the value in the bottom right cell (2 in the case in point, Figure 10):

•	•	k	a	s	t	e
•	0	1	2	3	4	5
k	1	0	1	2	3	4
a	2	1	0	1	2	3
r	3	2	1	1	2	3
e	4	3	2	2	2	**2**

Figure 10: finishing the computation

This method of computation is satisfying because it can detect invariant symbols even if they do not occupy the same positions in the two strings, like "e" in the example above.

For our research, however, it was necessary that the output of the computation corresponded to a transformation rather than to a single numeric value. A crucial point is that the distance measure is minimal and unique, but not so the set of transformations that brought to it. Let us examine the matrix in the example above; two different transformation sequences in (3) give the same distance measure, i.e. 2:

(3) N (k); N (a); S (r,s); D (t); N (e)
N (k); N (a); D (s); S (r,t); N (e)

To take another example, let us see the transformations of [sedevo] into [sjedo] (corresponding, respectively, to the 1SG of the imperfect and to the 1SG of the present of *sedere* 'sit'):

•	•	s	e	d	e	v	o
•	0	1	2	3	4	5	6
s	1	0	1	2	3	4	5
j	2	1	1	2	3	4	5
e	3	2	1	2	2	3	4
d	4	3	2	1	2	3	4
o	5	4	3	2	2	3	3

Figure 11: Levenshtein algorithm computed for *sedevo* and *siedo*

In this case, three possible paths give the result 3:

(4) a. N(s); D(e); S(d,j); N(e); S(v,d); N(o)
b. N(s); S(e,j); D(d); N(e); S(v,d); N(o)
c. N(s); I(j); N(e); N(d); D(e); D(v); N(o)

So, we developed a modified implementation of the Levenshtein algorithm, that gives exactly the output above. Additionally, the transformation paths can be simplified, by grouping together consecutive similar transformations:

(5) d. N(s); S(ed,j); N(e); S(v,d); N(o)
e. N(s);I(j);N(ed);D(ev);N(o)

Of course not all these transformation chains are morphologically significant, like for instance (5d) above. We will see how some of these are peculiar to single form pairs of single lexemes, while others are common to many lexemes.

Note, moreover, that the algorithm in question works promisingly also for non-concatenative inflection. For instance, given as input the two Arabic forms *faʕaltu* 'I did' and *saʔimtu* 'I hated', the algorithm gives the following output ("+" stands for the lexeme-variant matter, to be replaced by the lexeme's specific stem for that paradigm cell, and "/" suggests the discontinuity in the stems of Arabic verbs):

(6) faʕaltu → saʔimtu:
operations: S(f,s); N(a); S(ʕ,ʔ); S(a,i); S(l,m); N(t); N(u)
model: +a+tu
stems: f/ʕal, s/ʔim

4. Models and stems

By applying the algorithm described above to all lexemes contained in the corpus for each paradigm cell, we obtained the models in Table 4:

IND.PRES.1S	+
IND.PRES.2S	+
IND.PRES.3S	+
IND.PRES.1P	+'amo
IND.PRES.2P	+te
IND.PRES.3P	+no
IND.IMPF.1S	+o
IND.IMPF.2S	+i
IND.IMPF.3S	+a
IND.IMPF.1P	+v'amo
IND.IMPF.2P	+v'ate
IND.IMPF.3P	+ano
IND.PREM.1S	+
IND.PREM.2S	+sti
IND.PREM.3S	+
IND.PREM.1P	+mmo
IND.PREM.2P	+ste
IND.PREM.3P	+o
IND.FUT.1S	+r'ɔ

IND.FUT.2S	+r'aj
IND.FUT.3S	+r'a
IND.FUT.1P	+r'emo
IND.FUT.2P	+r'ete
IND.FUT.3P	+r'anno
CONG.PRES.1S	+
CONG.PRES.2S	+
CONG.PRES.3S	+
CONG.PRES.1P	+'amo
CONG.PRES.2P	+'ate
CONG.PRES.3P	+no
CONG.IMPF.1S	+ssi
CONG.IMPF.2S	+ssi
CONG.IMPF.3S	+sse
CONG.IMPF.1P	+ssimo
CONG.IMPF.2P	+ste
CONG.IMPF.3P	+ssero
COND.PRES.1S	+r'ɛj
COND.PRES.2S	+r'esti

COND.PRES.3S	+r'ɛbbe
COND.PRES.1P	+r'emmo
COND.PRES.2P	+r'este
COND.PRES.3P	+r'ɛbbero
IMP.2S	+
IMP.2P	+te
GERU	+ndo
INF	+re
PART.PRES.S	+nte
PART.PRES.P	+nti
PART.PASS.M.S	+o
PART.PASS.M.P	+i
PART.PASS.F.S	+a
PART.PASS.F.P	+e
D_TIVO.S	+'ivo
D_TORE.S	+'ore
D_ZIONE.S	+j'one

Table 4: invariants for each paradigm cell

Furthermore, it happens that for some paradigm cell sets, all the lexemes have the same stem, or the same stem set in case of overabundancy, like in the example below (Table 5):

cell	*model*	AMARE	CRESCERE	FARE
IND.IMPF.1P	+v'amo	ama	kreʃʃe	fatʃe
IND.IMPF.2P	+v'ate	ama	kreʃʃe	fatʃe

Table 5: consistent identity of stems for some groups of paradigm cells

Thus, only one paradigm cell from the set has to be included in further computations, and we will refer to that set as *metacell*.

5. Finding transformation rules

At this point, our interest lay in extracting a list of possible relations occurring between the stems of a lexeme. The modified Levenshtein Algorithm we implemented was applied to every pair of stems for all lexemes, each representing a metacell.

Table 6 shows the relations between the stem of the INDICATIVE PRESENT 1S and all the other stems (the figures indicate the number of lexemes for which the relation is valid):

CONG.PRES.1S	122 (87%)	N;S(a,o)	ABOLIRE, ACCENDERE, ADDURRE, …
CONG.PRES.1S	10 (7%)	N;S(i,o)	ABBANDONARE, AMARE, CONSUMARE, …
CONG.PRES.1S	2 (1%)	N;S(ia,ɔ)	DARE, STARE
CONG.PRES.1S	2 (1%)	N;S(i,jo)	CAMBIARE, COPIARE
CONG.PRES.1S	2 (1%)	N;D(j);N;S(da,ggo)	POSSEDERE, SEDERE
CONG.PRES.1S	2 (1%)	N;I(j);N;S(gga,do)	POSSEDERE, SEDERE
CONG.PRES.1S	1 (0%)	N;S(va,bbo)	DOVERE
CONG.PRES.1S	1 (0%)	N;S(ia,ono)	ESSERE
CONG.PRES.1S	1 (0%)	N;S(abbja,ɔ)	AVERE
CONG.PRES.1S	1 (0%)	N;S(bba,vo)	DOVERE
CONG.PRES.1S	1 (0%)	N;S(appja,ɔ)	SAPERE
IND.PRES.3P	120 (86%)	N	ABOLIRE, ACCENDERE, ADDURRE, AFFIGGERE, …
IND.PRES.3P	12 (8%)	N;S(a,o)	ABBANDONARE, AMARE, CAMBIARE, …
IND.PRES.3P	4 (2%)	N;S(an,ɔ)	AVERE, DARE, SAPERE, STARE
IND.PRES.3P	2 (1%)	N;D(j);N;S(d,gg);N	POSSEDERE, SEDERE
IND.PRES.3P	2 (1%)	N;I(j);N;S(gg,d);N	POSSEDERE, SEDERE

Table 6: relations of the IND.PRES.1S stem

Table 7 below shows the relations between the stem of the nominal derivative *-zione* and some other stems:

D_TIVO.S	11 (40%)	N;S(t,ʦ)	ABOLIRE, ASSOLVERE, ASSUMERE, DARE, …
D_TIVO.S	8 (29%)	N	ALLUDERE, CONCEDERE, DIFENDERE, EMETTERE, LEDERE, …
D_TIVO.S	7 (25%)	N;S(tt,ʦ)	ADDURRE, CONDURRE, COSTRUIRE, DIRIGERE, …
D_TIVO.S	1 (3%)	N;S(tt,ss)	CONNETTERE
D_TORE.S	11 (28%)	N;S(tt,ʦ)	ADDURRE, CONDURRE, COSTRUIRE, DIRIGERE, …
D_TORE.S	10 (26%)	N;S(t,ʦ)	APPARIRE, ASSOLVERE, ASSUMERE, CONSUMARE, …
D_TORE.S	7 (18%)	N	AGGREDIRE, CONCEDERE, DIFENDERE, FONDERE, …
(…)			
PART.PASS.M.S	9 (16%)	N;D(');N;S(t,ʦ)	ABOLIRE, ASSUMERE, CONSUMARE, …
PART.PASS.M.S	7 (12%)	N;D(');N;S(tt,ʦ)	AFFLIGGERE, CUOCERE, DISTRUGGERE, …
PART.PASS.M.S	7 (12%)	N;D(');N	AFFIGGERE, ALLUDERE, CONNETTERE, …
PART.PASS.M.S	3 (5%)	N;S('ɛtt,eʦ)	DIRIGERE, LEGGERE, PREDILIGERE
(…)			
IND.PRES.2P	7 (12%)	N;D(');N;I(ʦ)	ABOLIRE, APPARIRE, CONSUMARE, DARE, …
IND.PRES.2P	5 (8%)	N;S('a,aʦ)	CONSUMARE, DARE, FARE, MASTICARE, …
IND.PRES.2P	4 (7%)	N;S(d'e,s)	ACCENDERE, ARDERE, DIFENDERE, TENDERE
IND.PRES.2P	3 (5%)	N;S(d'e,ss)	CEDERE, CONCEDERE, POSSEDERE

Table 7: relations of the -ZIONE stem

6. Conclusion and future work

With this work, we researched an automatic method of identifying verbal stems and finding their morphophonetic relations. An ad-hoc implementation of the Levenshtein algorithm seems to give the expected results on a corpus of Italian verbal forms representing all the conjugation models. Moreover, the same procedure can be conducted by introducing to the corpus also the forms of deverbal derivatives, in order to include them in the paradigmatic analysis of inflection.

Future work should undoubtedly include a method for constructing chains of predictability, based on the computation of the conditional entropy of the relations found, coupled with frequency data from a corpus.

References

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

Bonami, O., Boyé, G. & F. Kerleroux (2009) L'allomorphie radicale et la relation flexion-construction. In: B. Fradin, F. Kerleroux, & M. Plénat (Eds.), *Aperçus de morphologie du français*. Saint-Denis: Presses Universitaires de Vincennes, 103-125.

De Mauro, T. (Ed.) (1999) *Grande dizionario italiano dell'uso*. Torino: Utet.

Hamming, R. W. (1950) Error Detecting and Error Correcting Codes. *Bell System Technical Journal* 29 (2): 147-160.

Levenshtein, V. I. (1966) Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10: 707-710.

Maiden, M. (1992) Irregularity as a Determinant of Morphological Change. *Journal of Linguistics* 28(2): 285-312.

Montermini, F. & G. Boyé (2012) Stem relations and inflection class assignment in Italian. *Word Structure* 5(1): 69-87.

Pirrelli, V. & M. Battista (2000) The Paradigmatic Dimension of Stem Allomorphy in Italian Verb Inflection. *Italian Journal of Linguistics* 12: 307-379.

Stump, G. T. (2001) *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge: Cambridge University Press.