Inflectional morphology with linear mappings

R. Harald Baayen[1], Yu-Ying Chuang[1], and James P. Blevins[2]
[1] University of Tübingen | [2] University of Cambridge

This methodological study provides a step-by-step introduction to a computational implementation of word and paradigm morphology using linear mappings between vector spaces for form and meaning. Taking as starting point the linear regression model, the main concepts underlying linear mappings are introduced and illustrated with R code. It is then shown how vector spaces can be set up for Latin verb conjugations, using 672 inflected variants of two verbs each from the four main conjugation classes. It turns out that mappings from form to meaning (comprehension), and from meaning to form (production) can be carried out loss-free. This study concludes with a demonstration that when the graph of triphones, the units that underlie the form space, is mapped onto a 2-dimensional space with a self-organising algorithm from physics (graphopt), morphological functions show topological clustering, even though morphemic units do not play any role whatsoever in the model. It follows, first, that evidence for morphemes emerging from experimental studies using, for instance, fMRI, to localize morphemes in the brain, does not guarantee the existence of morphemes in the brain, and second, that potential topological organization of morphological form in the cortex may depend to a high degree on the morphological system of a language.

Keywords: inflectional morphology, linear discriminative learning, word and paradigm morphology, linear mappings, 2-D network topology, graphopt

1. Introduction

Introductions to linguistics and textbooks on morphology frame the discussion of form variation in terms of the theoretical construct of the morpheme, defined as the smallest linguistic unit combining form and meaning (see, e.g., Spencer, 1991; Plag, 2003; Booij, 2012; Lardier, 2014). Unsurprisingly, the vast majority of studies

https://doi.org/10.1075/ml.18010.baa | Published online: 10 January 2019
The Mental Lexicon 13:2 (2018), pp. 230–268. issn 1871-1340 | e-issn 1871-1375

on morphological processing assumes that morphemes are well-established theoretical notions and also exist in the mind. The dominant view of the mental lexicon in psychology and cognitive science is well represented by Zwitserlood (2018, p. 583):

> Parsing and composition – for which there is ample evidence from many languages – require morphemes to be stored, in addition to information as to how morphemes are combined, or to whole-word representations specifying the combination.

as well as by Butz and Kutter (2016, Chapter 13), according to whom “morphemes are the smallest meaning bearing grammatical units”. Words are believed to be built from morphemes, and the meanings of complex words are assumed to be a compositional function of the meanings of their parts. In a morpheme-based lexicon, an agent noun such as *worker* can be derived by unification of the representations of its constituents (Lieber, 1980). Thus, given the verb *work*

< work; [V];WORK) >

and the derivational morpheme

< -er; [N|V];AGENT() >

the nominalization

< work-er; [[[V][N|V]]N]; < AGENT(WORK) >>

is obtained. Reflecting a longstanding confound, the term ‘morpheme’ is often used in experimental work on the mental lexicon, to refer solely to the form part of the linguistic morpheme (i.e., the ‘morph’). In comprehension, accessing this form is seen as essential for gaining access to its meaning and its combinatorial properties (Taft, 1994; Marantz, 2013). In production, given the semantics to be expressed, a morphemic form is selected and aligned with the form of its base to prepare for articulation (Levelt et al., 1999).

However, the theoretical construct of the morpheme as the smallest unit combining form and meaning is highly problematic. Aronoff and Fudeman (2011) explicitly back off from the idea that morphemes are linguistic signs.

> We have purposely chosen not to use this definition. Some morphemes have no concrete form or no continuous form ... and some do not have meanings in the conventional sense of the term.

Beard (1977) observed out that in language change, morphological form and morphological meaning follow their own trajectories, and as a consequence the theoretical construct of the morpheme as a minimal sign combining form and meaning stands in the way of understanding the temporal dynamics of language change. Matthews, in his introduction to morphology (Matthews, 1974, 1991), pointed out that the inflectional system of a language such as Latin is not well served by analyses positing that its fusional system is best analyzed as underlyingly agglutinative (see also Hockett, 1960). One striking example of the non-compositional nature of inflectional morphology is provided by Estonian. In this language, most of the plural forms of nouns are built starting with the form of the partitive singular, without the semantics of these plural forms expressing in any way the semantics of the singular and the partitive (e.g., for JALG 'foot', *jalga,* partitive singular, *jalgadele,* adessive plural, see Erelt, 2003; Blevins, 2006).

A word-based perspective is common to a range of morphological approaches. As Booij (2018) clarifies, construction morphology is

> … word-based morphology. That is, complex words are not seen primarily as a concatenation morphemes, but as independent meaningful units within which certain subcomponents (morphemes) may be distinguished on the basis of paradigmatic relations with other words. That is, morphology is not to be equated with the 'syntax of morphemes'. Morphological schemas characterize the 'Gestalt' of complex words and their holistic properties. (p. 4–5)

In what follows, we restrict the use of the term *morpheme* to the concept of the morpheme as a minimal linguistic sign combining form and meaning in a syntax of minimal signs and combinations thereof, a concept that goes back to post-Bloomfieldian American structuralism (Blevins, 2016). A word-based perspective is fully compatible with notions of word-internal structure. We will use the term *exponent* (a term that goes back to Matthews) to refer to recurrent units of subword form (or form variation). These subcomponents or variants correlate with paradigmatic relations between words. Exponents are often understood in realizational models as the phonological 'spell-out' of one or more morphosyntactic properties (Stump, 2001), but they can also be interpreted discriminatively, as markers that distinguish larger meaningful units.

Although realizational theories eschew the construct of the morpheme, they embrace the construct of the exponent. For the most part, realizational models have focused on patterns of inflectional exponence. For example, the rule proposed in Matthews (1991, p. 127) spells out regular plurals in English by concatenating [z] to the noun stem *x*.

(1) $< [\text{Plural}, x]_N \rightarrow x + [\text{z}] >$

The notion of an exponent as a 'spell-out' or 'marker' can also be extended to the derivational domain. The constructional schema proposed by Booij (2016) (Example 3) illustrates this for English deverbal agent nouns,

(2) $< [x]\mathrm{v}_{i}\ \mathrm{er}]\mathrm{N}_{j} \leftrightarrow [\text{Agent of SEM}_{i}\,]_{j} >$

where the verbal stem *x* is followed by the exponent *er*. For derived words with their own shades of meanings, further subschemata may be required. For *worker,* http://www.dictionary.com/ lists several meanings, including (1) a person or thing that works, (2) a laborer or employee, (3) a person engaged in a particular field, activity, or cause, or (4) a member of a caste of sexually underdeveloped, nonreproductive bees, specialized to collect food and maintain the hive. Schema (2) is adequate for the first reading. The other meanings require additional subschemata. For the worker-bee, such a subschema could be:

(3) $< [x]\mathrm{v}_{i}\ \mathrm{er}]\mathrm{N}_{j} \leftrightarrow [\text{IS\_BEE}[\text{Agent of SEM}_{i}\,]_{j}\,] >$

Using the computational mechanism of inheritance hierarchies, the idiosyncratic part of the lexical entry for worker-bee, [IS_BEE[]] can be separated out from the more general agentive nominalization defined by (2). Inheritance hierarchies have been applied most systematically in approaches that aim to factor variation into patterns of varying levels of generality, from patterns that characterize whole word classes, through those that distinguish inflection classes or subclasses, down to lexical idiosyncrasies. The most highly developed of these approaches include models of Network Morphology (Corbett and Fraser, 1993; Brown and Hippisley, 2012) and allied DATR-based accounts (Cahill and Gazdar, 1999).

For constructional analyses of inflectional paradigms, second-order schemata can be set up. The second-order schema formulated by Booij (2016) (Example 39) for English singular and plural nouns,

(4) $< [(x_{i}\,)_{\omega\text{-}j} \leftrightarrow [\mathrm{N}_{i}, +\mathrm{sg}]_{j} \leftrightarrow [\text{SING}[\text{SEM}_{i}\,]]_{j} > \approx < [(x_{i}\,\text{-z})_{\omega\text{-}j} \leftrightarrow [\mathrm{N}_{i}, +\mathrm{pl}]_{j} \leftrightarrow$
$[\text{PLU}[\text{SEM}_{i}\,]]_{j} >$

illustrates that whereas a fully compositional calculus is set up for the meanings of complex words (cf. Jackendoff, 1990), the forms of these words are not hierarchically structured. Thus, an exponent realizing one or more semantic functions is nothing more than an additive change to the phonological form of a word.

At a high level of symbolic abstraction, realizational and constructional theories isolate variation in form and meaning and clarify how form and meaning can go together in complex words. This type of analysis may be of practical value, especially in the context of adult second language acquisition. It is less clear whether the corresponding theories, whose practical utility derives ultimately from their pedagogical origins, can be accorded any cognitive plausibility. Constructional schemata, inheritance, and mechanisms spelling out exponents are all products of descriptive traditions that evolved without any influence from research traditions in psychology. As a consequence, it is not self-evident that these notions would provide an adequate characterization of the representations and processes underlying comprehension and production. It seems particularly implausible that children would be motivated to replicate the descriptive scaffolding of theoretical accounts and attempt to establish the systems of 'inflection classes' proposed for languages such as Estonian or the 'templates' associated with languages like Navajo.

The representations in current morphological theories and descriptions also tend to gloss over the actual details of words' forms as they are used in daily life. In spontaneous conversational speech, words can have substantially reduced forms (Ernestus et al., 2002; Johnson, 2004; Kemps et al., 2004). Furthermore, the actual phonetic realization of an exponent may depend on the semantic function that it realizes (Plag et al., 2017). Another complication is that the meanings of words, as illustrated above for *worker,* can be much richer than specified by the semantic function in constructional schemas such as (2). It therefore makes sense for Booij (2018) to characterize morphological schemas as the 'Gestalt' of complex words, but this raises new questions about the extent to which a schema such as (3) is truly transparent to the more general schema (2) of which it is an idiosyncratic instantiation.

The conception of words as 'recombinant Gestalts' (Ackerman et al., 2009) highlights a further difficulty. Realizational models retreat from the static form-meaning correspondence encapsulated in the structuralist morpheme. However, they retain a dynamic counterpart in assuming a stable correlation between the differences in meaning and differences in form between words. In the case of regular English plurals, the difference between the presence or absence of the 'plural' feature correlates with the presence or absence of the marker [z]. However, Gestalts do not work in this way. For example, the partitive singular of JALG 'foot' cited above, *jalga,* contrasts with the nominative singular *jalg.* Yet the contrast between partitive and nominative singular case is not realized by the presence or absence of -*a*. Although -*a* discriminates partitive and nominative singular forms, it is the theme vowel of jalg, and occurs in the genitive singular and nearly all forms of the noun.

A further problem is that morpho-syntactic features are not necessarily tied to a specific semantic function. For instance, ablative case in Latin can realize semantic functions that in English would be expressed by prepositions as different as *from, with, by, in,* and *at.* The reason that exponents such as *o* (for a particular class of masculine nouns) and *ae* (for a particular class of feminine nouns) are analyzed as realizing the ablative is that words with these exponents occur with the same range of abovementioned prepositional meanings. In other words, the morphosyntactic feature 'ablative' does not represent a semantic function, but a distribution class. This brings us to Word and Paradigm Morphology.

The approach of Word and Paradigm Morphology (Matthews, 1974, 1991) is different from that of construction morphology, in that proportional analogies between words within paradigms are assumed to make the lexicon as a system productive. As explained by Matthews (1991),

> In effect, we are predicting the inflections of servus by analogy with those of dominus. As Genitive Singular domini is to Nominative Singular dominus, so x (unknown) must be to Nominative Singular servus. What then is x? Answer: it must be servi. In notation, dominus : domini = servus : servi. (p. 192f)

Only words exist, and an exponent such as the *-i* that realizes the genitive singular for masculine (and neuter) nouns of particular declension classes is implicit in the paradigmatic relations that characterize the Latin noun system. Importantly, exponents themselves have no independent existence, at best, they are a descriptive device useful to highlight paradigmatic analogies between the only units that do exist: full words.

An important aspect of Word and Paradigm Morphology is that morpho-syntactic features in general represent distribution classes (Blevins, 2003, 2016), as explained above for the Latin ablative. But the same pertains to, for instance, the English singular, the use of which includes reference to single instances for count nouns (*the pen),* but also reference to non-individuated quantities (*the milk*), categories (*the evil),* and organizations (*the church).* The similarity in meaning of genitive or ablative forms in Latin, or the singular in English, then follows from the distributional hypothesis, which proposes that linguistic forms with similar distributions have similar meanings (Weaver, 1955; Firth, 1968; Landauer and Dumais, 1997). Thus, Matthew's analogy of forms

dominus : domini = servus : servi.

is actually paralleled by an analogy of lexical co-occurrence distributions d:

$d(\text{dominus}) : d(\text{domini}) = d(\text{servus}) : d(\text{servi})$.

We can highlight the complex system of multidimensional analogies at issue by writing

$$(5)\quad \begin{pmatrix} \text{dominus} & : & d(\text{dominus}) & = & \text{servus} & : & d(\text{servus}) \\ : & & : & & : & & : \\ \text{domini} & : & d(\text{domini}) & = & \text{servi} & : & d(\text{servi}) \end{pmatrix}.$$

However, for this perspective on inflectional morphology and paradigmatic analogy to become computationally tractable, it is essential to (1) define words' forms in such a way that similarities between words can be calculated, (2) to represent the meanings of words in a distributional way so that the semantic similarity between words can be quantified, and (3) to formalize paradigmatic analogy mathematically so that it becomes computationally tractable. The theory laid out in Baayen et al. (2018) provides a simple but effective implementation of these ideas.

Word forms are represented by vectors of zeroes and ones specifying which triphones make up these forms. The choice for triphones is motivated by two considerations. First, phones are inherently contextual; for instance, information about the place of articulation of plosive is carried by formant transitions in adjacent vowels. Second, triphones encode information about partial order: evidence for the triphones han and and implies a directed path han → and. For languages with strong phonotactic constraints, diphones are expected to be more effective as sublexical units of form (Pham and Baayen, 2015). For a richly inflecting language such as Latin, it is conceivable that four-phone sequences are also effective sublexical units. Given form units such as triphones, each word form is conceptualized as a point in a high-dimensional form space. How similar two forms are can be evaluated straightforwardly with the Pearson correlation of the pertinent form vectors.

Word meanings are also represented by numeric vectors, taking inspiraton from computational models of distributional semantics (see, e.g., Landauer and Dumais, 1997; Shaoul and Westbury, 2010; Mikolov et al., 2013). In what follows, we use the Pearson correlation to evaluate the similarity between semantic vectors. Another commonly used measure is the cosine similarity. Which measure is optimal depends in part on how semantic vectors are calculated. For the present study, the correlation measure turns out to be effective and is therefore selected.

Given vector representations for forms and meanings, we can now evaluate mathematically (and computationally) how similar words are and how similar meanings are. In other words, we can evaluate the similarities between the representations on the top row of (5) and the corresponding representations on the bottom row. What is still missing is a way of evaluating analogies within the rows of (5). For this, Baayen et al. (2018) propose to use the linear mapping from linear algebra. They show that for English, their Linear Discriminative Learner (LDL) model, which implements such mappings, is remarkably succesful for the computational modeling of basic inflectional and derivational morphology.

The present study extends the LDL model to a language with a much richer inflectional system, Latin. We constructed a dataset that contains, for 8 verbs selected from the four major conjugation classes of Latin, all of the 6 forms in each of 14 paradigms. For each of the 672 paradigm forms in this dataset, we constructed the corresponding form vectors. We also constructed semantic vectors. Baayen et al. (2018) derived semantic vectors for English from a corpus.

In the present study, we simulated semantic vectors, leaving the construction of real semantic vectors from corpora of Latin to future research.

In what follows we pursue three goals. First, we seek to clarify, by means of a case study of a non-trivial inflectional system, whether it is actually possible for LDL to produce and understand inflected forms computationally without having to hand-craft stems, exponents, inflection classes, and inheritance hierarchies.

Our second goal is to provide an accessible introduction to the basic mathematical concepts used in LDL, and to provide a step-by-step guide to how this model can be applied using the statistical programming environment R (R Core Team, 2017). A package for R that implements Word and Paradigm Morphology with Linear Discriminative Learning (**WpmWithLdl**) can be downloaded from http://www.sfs.uni-tuebingen.de/~hbaayen/publications/WpmWithLdl_1.0.tar.gz.

Our third goal is to reflect on the cognitive plausibility of form units such as morphemes and exponents. A vast majority of experimental studies interprets effects observed in the laboratory as evidence for the psychological reality of the morpheme (see, e.g., Zwitserlood, 2018). However, reasoning from the consequent to the antecedent confuses necessity with sufficiency. We illustrate this fallacy for a study reporting the supposed localization of morphemes in the brain (Bozic et al., 2010) by showing that our model predicts topological clustering of form that would seem to support the existence of morphemes – even though there are no morphemes or exponents whatsoever in our model. The hypothesis that we will put forward is that experimental effects traditionally understood as evidence for morphemes (or exponents) can be explained just as well with LDL.

Before introducing the Latin dataset, let us briefly clarify the similarities and differences between LDL and realizational/constructional theories of morphology (R/CM). Both LDL and R/CM assume that

1. morphology is word-based,
2. a word has at least one form and at least one meaning,
3. word forms do not have an internal hierarchical structure (no syntax of morphemes),
4. the meaning of an inflected or derived word incorporates 'grammatical' meanings such as agent or plural,
5. especially derived words can have their own idiosyncratic semantics, and
6. the system is productive.

LDL and R/CM differ in the following respects:

1. In R/CM, the word forms of complex words are constructed from persistent stems and exponents, whereas in LDL, there are no units for stems nor for exponents.

2. In R/CM, meanings tend to be tacitly associated with grammatical features. When specified, these meanings often take the form of monadic units, such as SEM $_i$ in (2), or compositional functions, such as PLU[] in (4); in LDL, all meanings are represented by semantic vectors, importantly, inflectional and derivational meanings are represented by their own specific semantic vectors.
3. In R/CM, the meaning of a complex inflected word is determined by the features associated with the word. The meaning of a complex derived word is assumed to be a compositional function of the meanings of its parts. LDL, by contrast, is a discriminative theory in which the meaning of a transparent complex word is obtained by integrating over the semantic vectors of the base word and the semantic vector of the inflectional or derivational function (see Baayen et al., 2018, for detailed discussion). The semantic vector of a derived word, based on its own distribution, can differ from that obtained by integrating over the semantic vectors of stem and derivational function. The greater this difference, the more the derived word is perceived as semantically opaque (see also Marelli and Baroni, 2015).
4. In R/CM, representations and operations (such as inheritance) on representations are agnostic with respect to the tasks of production and comprehension. In LDL, production and comprehension have their own specific mappings (which mathematically are (approximately) each other's inverse).

In summary, LDL is a computational implementation of Word and Paradigm Morphology in which analogy, formalized as linear mappings over vectorized representations of form and meaning, drives comprehension and production. In this approach it is no longer necessary to hand-craft forms for stems and exponents, or to set up inflectional classes and inheritance hierarchies. LDL appears to work well for English (Baayen et al., 2018). But does it also work for more complex inflectional systems such as the conjugations of the Latin verb? As a first step towards addressing this question, we first introduce our dataset of Latin verbs.

2. Latin conjugations

The dataset on Latin conjugations is available in the `R` package **WpmWithLdl.** To install the package, assuming the package source (`WpmWithLdl_1.0.tar.gz`) is available in the directory visible to `R`, proceed as follows:

```
install.packages("WpmWithLdl_1.0.tar.gz", repos=NULL)
```

The Latin dataset is available for use after executing the following commands:

```
library(WpmWithLdl)
data(latin)
```

The words under examination in this study are the finite verb forms of two verbs from each of the four major conjugation classes of Latin. For each verb, 14 paradigms (each with 3 persons × 2 numbers) were included (present/past × active/passive × indicative/subjunctive; perfect/pluperfect × indicative/subjunctive; future × active/passive). Table 1 presents the 14 inflected forms for the first person singular of the verb *vocaare, terrere, carpere,* and *audire,* each belonging to one of the four conjugation categories. Long vowels are represented by vowel doubling. It can be seen that each conjugation class has its own idiosyncracies. For instance, the first person singular of the 1st conjugation class does not contain the theme vowel characteristic for this class; the 2nd conjugation class has a perfect form without the v exponent found in the 1st and 4th classes; the 3rd conjugation class has a different stem form for the perfect tenses; and the 3rd and 4th conjugation classes do not make use of the b exponent for the future as used in the 1st and 2nd conjugation classes. In total, there are 672 different verb forms in the dataset.

Table 1. The 14 inflected forms for the first person singular of the verbs *vocaare, terrere, carpere,* and *audire,* each of which represents one of the four conjugation classes

CLASS I	CLASS II	CLASS III	CLASS IV	tense	voice	mood
vocoo	terreoo	carpoo	audioo	present	active	ind
vocem	terream	carpiam	audiam	present	active	subj
vocor	terreor	carpor	audior	present	passive	ind
vocer	terrear	carpiar	audiar	present	passive	subj
vocaaboo	terreeboo	carpam	audiam	future	active	ind
vocaabor	terreebor	carpar	audiar	future	passive	ind
vocaabam	terreebam	carpeebam	audieebam	past	active	ind
vocaarem	terreerem	carperem	audiirem	past	active	subj
vocaabar	terreebar	carpeebar	audieebar	past	passive	ind
vocaarer	terreerer	carperer	audiirer	past	passive	subj
vocaavii	terruii	carpsii	audiivii	perfect	active	ind
vocaaverim	terruerim	carpserim	audiiverim	perfect	active	subj
vocaaveram	terrueram	carpseram	audiiveram	pluperfect	active	ind
vocaavissem	terruissem	carpsissem	audiivissem	pluperfect	active	subj

The task that we set ourselves is to find a mapping from the Latin verb forms onto their meanings, and also a mapping from these meanings onto the corresponding forms, without making use of morphemes, exponents, and stems, and without having to set up inflectional classes. A trivial and uninteresting mapping would be to pair each of the 672 word forms with a monadic form unit, indexed from $i = 1, 2, \ldots, 672$, and to set up a second set of monadic semantic units, indexed from $j = 1$, 2,..., 672. We can then define pairs (i, j) and a function $f(i)=j$ as well as a function $g(j)=i$ that produce the meaning unit given a form unit, and a form unit, given the meaning unit of a pair, respectively. Such a full listing set-up is uninteresting as it does not do justice to the similarities between forms, the similarities between meanings, the analogies between forms and meanings, and the productivity of the system.

The approach that we adopt in what follows is to pair each word form i with a numeric vector $\boldsymbol{c}_i$ characterizing its form and a second numeric vector $\mathbf{s}_i$ characterizing its meaning. Below, we explain how form vectors and semantic vectors can be set up. Here, we note that the word forms can now be conceptualized as points in a high dimensional form space $\{C\}$, and that word meanings can likewise be conceptualized as points in a second high-dimensional space $\{S\}$. We are interested in a mapping $\boldsymbol{F}$ that takes the points in $\{C\}$ as input and produces the points in $\{S\}$ as output. This mapping represents the comprehension of inflected forms. Likewise, we can set up a mapping from $\{S\}$ to $\{C\}$ to represent the production of inflected forms. In this study, we constrain these mappings to be linear. In what follows, we first introduce the mathematics of linear mappings. We then return to Latin and show how a form space and a semantic space can be constructed. We then examine how succesful linear mappings are for moving back and forth between form and meaning.

3. Introduction to the mathematics of linear mappings

Anyone who has run a linear regression analysis has made use of a particular instantiation of a linear mapping. Consider a dataset with n observations, each consisting of a response y_i and k predictors $x_{i1}, x_{i2}, \ldots, x_{ik}$. By way of example, the response could be reaction time to words in a lexical decision task, and the predictors could be frequency of occurrence, word length, number of neighbors, ... A data table for such a data set with $k=2$ predictors is of the form:

$$\begin{array}{ccc} y_1 & x_{11} & x_{12} \\ y_2 & x_{21} & X_{22} \\ \vdots & \vdots & \vdots \\ y_n & X_{n1} & X_{n2} \end{array}$$

The estimates $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$ predicted by a linear regression model for observations $i=1, 2, \ldots n$ are a weighted sum of the data points $x_{i1}, x_{i2}, \ldots, x_{ik}$ $(i=1, 2, \ldots, n)$, with weights $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k$ estimated from the data. For the above dataset with two predictors $(k=2)$, we can write

(6)
$$\begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{pmatrix} = \begin{pmatrix} \hat{\beta}_0 & + & \hat{\beta}_1 x_{11} & + & \hat{\beta}_2 x_{12} \\ \hat{\beta}_0 & + & \hat{\beta}_1 x_{21} & + & \hat{\beta}_2 x_{22} \\ \vdots & + & \vdots & + & \vdots \\ \hat{\beta}_0 & + & \hat{\beta}_1 x_{n1} & + & \hat{\beta}_2 x_{n2} \end{pmatrix},$$

Here, $\hat{\beta}_0$ represents the estimated intercept, and $\hat{\beta}_1$ and $\hat{\beta}_2$ the estimated slopes for the predictors $x._1$ and $x._2$ respectively. Using notation from linear algebra, we can restate this equation more succintly as follows. First, let

$$\hat{y} = \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{pmatrix},$$

let

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} \end{pmatrix},$$

and let

$$\hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix}.$$

We use lower case bold font to denote column vectors, and upper case bold font to denote matrices (bundles of column vectors). The column vector with sums on the right hand side of (6) is the product of $\boldsymbol{X}$ and $\hat{\beta}$. How to calculate the product of two matrices is illustrated for two 2×2 matrices in Figure 1. To obtain the value in the cell in the upper right of the resulting matrix (c_{12}), one takes the first row of the first matrix and the second column of the second matrix. These two vectors are aligned, the values are pairwise multiplied, and the resulting products are summed. The same procedure generalizes to larger matrices. In general, given matrices $\boldsymbol{A}$ (r rows and s columns) and $\boldsymbol{B}$ (s rows and t columns), the product $\boldsymbol{C} = \boldsymbol{AB}$ is an $r \times t$ matrix such that the element in the i-th row and j-th column of $\boldsymbol{C}$ is given by

$$c_{ij} = \sum_{k=1}^{s} a_{ik} b_{kj}.$$

Thus, for the product of two matrices to be defined, the number of columns of the first matrix should be the same as the number of rows of the second matrix. In the case of the regression weights $\hat{\beta}$, the number of rows of the 3×1 column matrix $\hat{\beta}$ equals the number of columns of $\boldsymbol{X}$. Hence, (6) can be rewritten as

(7) $\hat{\boldsymbol{y}}=\boldsymbol{X}\hat{\boldsymbol{\beta}}$.

In standard linear regression, we have one vector of observations, $\boldsymbol{y}$. However, we can consider multiple response vectors simultaneously, which we bring together by placing their vectors into the rows of a matrix $\boldsymbol{Y}$. We can now generalize (7) to obtain the following model:

(8) $\hat{\boldsymbol{Y}}=\boldsymbol{X}\hat{\boldsymbol{B}}$,

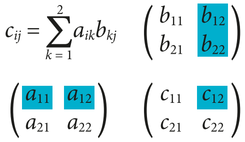
$c_{ij} = \sum_{k=1}^{2} a_{ik}b_{kj}$
$\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$
$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$
$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$

Figure 1. Matrix multiplication illustrated for 2×2 matrices: c_{12} is $a_{11}b_{12}+a_{12}b_{22}$

with $\hat{\boldsymbol{B}}$ the matrix with the estimated regression weights. Geometrically, we can think of (8) as describing a mapping from one set of points in an *n*-dimensional space {*X*}, each point given by a row vector of $\boldsymbol{X}$, onto another *n*-dimensional space {*Y*} in which each point is given by a row vector of $\boldsymbol{Y}$. Ideally, $\boldsymbol{B}$ maps each point in {*X*} exactly onto its corresponding point in {*Y*}, but in practice, the mapping will often be approximate only. As in standard linear regression, we have to estimate the mapping $\boldsymbol{B}$ from the data. Given $\boldsymbol{B}$, we can then obtain an estimate $\hat{\boldsymbol{Y}}$ of $\boldsymbol{Y}$, which provides the estimated locations of the data points in {*Y*}.

Figure 2 illustrates the geometry of such a mapping for two spaces {*C*} and {*S*}, and two mappings, $\boldsymbol{F}$ and $\boldsymbol{G}$. The matrix $\boldsymbol{C}$ defines three datapoints in {*C*}, *a*, *b*, and *c*, with coordinates (1, 2), (−2, −2), and (−2, 1) respectively (shown in blue in the left part of Figure 2). The transformation matrix $\boldsymbol{F}$ maps these points onto the points (2, −4), (−4, 4), and (−4, −2) shown in red in the right part of this figure.

From the perspective of standard regression, $\boldsymbol{C}$ comprises two predictors, the *x*-coordinate and the *y*-coordinate. Each data point specifies values for these coordinates, and thus represents points in the plane spanned by the *x* and *y* axes, as shown in the left part of Figure 2. The first column of the matrix $\boldsymbol{F}$, which maps points in {*C*} onto points in {*S*}, specifies the regression weights that we need to obtain the predicted *x*-coordinates in the space {*S*}, and the second column of $\boldsymbol{F}$ likewise provides the regression weights for the predicted *y*-coordinates in {*S*}.

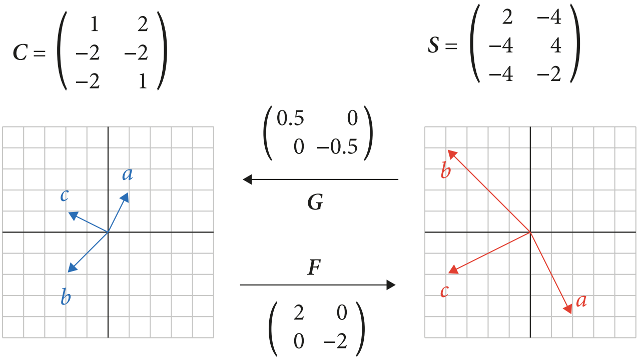
$C = \begin{pmatrix} 1 & 2 \\ -2 & -2 \\ -2 & 1 \end{pmatrix}$
$S = \begin{pmatrix} 2 & -4 \\ -4 & 4 \\ -4 & -2 \end{pmatrix}$
$\begin{pmatrix} 0.5 & 0 \\ 0 & -0.5 \end{pmatrix}$
G
F
$\begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$
a
c
b
b
c
a

Figure 2. Matrix ***F*** maps the row vectors of ***C*** onto the row vectors ***S*** ; conversely, matrix ***G*** maps the row vectors ***S*** back onto those of ***C***

An important property of a mapping from space {X} to space {Y} using some transformation matrix ***F*** is that there is a reverse mapping (which may be approximate only) from space {*Y*} back onto space {*X*} with a second transformation matrix, denoted by ***G*** in Figure 2. Just as matrix ***F*** maps points in {C} onto points in {*S*},

$$\boldsymbol{CF} = \boldsymbol{S},$$

matrix ***G*** maps points in {*S*} back onto points in {*C*}:

$$\boldsymbol{SG} = \boldsymbol{C}\,.$$

The following code snippet in R illustrates these mappings; **%*%** is R's operator for matrix multiplication.

```
C = matrix(c(1, -2, -2, 2, -2, 1), nrow = 3, ncol = 2)
S = matrix(c(2, -4, -4, -4, 4, -2), nrow = 3, ncol = 2)
F = matrix(c( 2, 0, 0, -2), nrow = 2, ncol = 2)
G = matrix(c(0.5, 0, 0, 0.5), nrow = 2, ncol = 2)
C%*%F
```

```
     [,1] [,2]
[1,]    2   -4
[2,]   -4    4
[3,]   -4   -2
```

```
S%*%G
```

```
     [,1] [,2]
[1,]    1   -2
[2,]   -2    2
[3,]   -2   -1
```

Given the points in spaces $\{C\}$ and $\{S\}$, the question arises of how to obtain the transformation matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ that map points from one space onto the other. For this, we require the matrix inverse. The inverse of a 2×2 square matrix $\boldsymbol{A}$, denoted by $\boldsymbol{A}^{-1}$, is defined as follows:

$$A^{-1} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}^{-1} = \frac{1}{a_{11}a_{22} - a_{21}a_{12}} \begin{pmatrix} a_{22} & -a_{21} \\ -a_{12} & a_{11} \end{pmatrix}.$$

If the determinant $a_{11}a_{22} - a_{21}a_{12}$ is zero (i.e., when $a_{11}a_{22} = a_{21}a_{12}$), the inverse is not defined, and $\boldsymbol{A}$ is referred to as a singular matrix. In `R`, the inverse of a non-singular, square matrix is obtained with `solve`:

```
solve(F)   # gives G

     [,1] [,2]
[1,]  0.5  0.0
[2,]  0.0 -0.5

solve(G)   # gives F

     [,1] [,2]
[1,]    2    0
[2,]    0    2
```

The product $\boldsymbol{A}^{-1}\boldsymbol{A}$ of a square matrix $\boldsymbol{A}$ is the identity matrix $\boldsymbol{I}$, i.e., a matrix with 1 on the main diagonal and 0 elsewhere. For instance,

```
solve(F) %*% F

     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

The matrices $\boldsymbol{C}$ and $\boldsymbol{S}$ are not square matrices, so we cannot use **solve** to obtain their inverse. We therefore derive the transformation matrix with a small detour, as follows.

(9)
$$\begin{aligned} \boldsymbol{C}\,\boldsymbol{F} &= \boldsymbol{S} \\ \boldsymbol{C}^{T}\boldsymbol{C}\boldsymbol{F} &= \boldsymbol{C}^{T}\boldsymbol{S} \\ (\boldsymbol{C}^{T}\boldsymbol{C})^{-1}\boldsymbol{C}^{T}\boldsymbol{C}\boldsymbol{F} &= (\boldsymbol{C}^{T}\boldsymbol{C})^{-1}\boldsymbol{C}^{T}\boldsymbol{S} \\ \boldsymbol{I}\boldsymbol{F} &= (\boldsymbol{C}^{T}\boldsymbol{C})^{-1}\mathrm{C}^{T}\mathrm{S} \\ \boldsymbol{F} &= (\boldsymbol{C}^{T}\boldsymbol{C})^{-1}\mathrm{C}^{T}\mathrm{S} \end{aligned}$$

In this derivation, we pre-multiply $\boldsymbol{C}$ with its transpose because this results in a matrix that is square and smaller in size,

```
t(C) %*% C

     [,1] [,2]
[1,]    9    4
[2,]    4    9
```

and because it is square, we can now apply optimized algorithms as implemented in `solve` to obtain its inverse:

```
solve(t(C) %*% C)

            [,1]        [,2]
[1,]  0.13846154 -0.06153846
[2,] -0.06153846  0.13846154
```

By multiplying $\mathbf{C}^{T}\mathbf{C}$ with its inverse, the left hand side of (9) simplifies to the transformation matrix $\boldsymbol{F}$, which is found to be equal to $(\mathbf{C}^{T}\boldsymbol{C})^{-1}\boldsymbol{C}^{T}\boldsymbol{S}$. When working with large matrices in which rows can be similar – which is often the case for language, since words can be similar in form or similar in meaning – it is often not possible to use **solve,** as the matrix can be too close to singular. In practice, we therefore use the Moore-Penrose generalized inverse, which is implemented in R as the **ginv** function of the **MASS** package. The generalized inverse of $\boldsymbol{X}$ is denoted as $\boldsymbol{X}'$.

Once $\boldsymbol{F}$ has been estimated, the predicted data points in $\{S\}$ are given by $\hat{\mathbf{S}}=\mathbf{CF}$. We can rewrite this equation as follows:

(10)
$$\begin{aligned} \hat{\boldsymbol{S}} &= \boldsymbol{CF} \\ &= \boldsymbol{C}(\boldsymbol{C}^{T}\boldsymbol{C})^{-1}\boldsymbol{C}^{T}\mathrm{S} \\ &= \boldsymbol{HS}, \end{aligned}$$

where $\boldsymbol{H}$ is the so-called hat matrix of the linear regression model.

In what follows, $\boldsymbol{C}$ denotes an $n \times k$ matrix describing the form properties of n words, with cell C_{ij} taking on the value 1 if word form i contains triphone j, and 0 otherwise. Thus, words are observations in a k-dimensional triphone space, and triphones are predictors for words. Furthermore, we will use $\boldsymbol{S}$ to denote an $n \times k$ matrix the row vectors of which are real-valued semantic vectors of length k. Ideally, these vectors are estimated from corpora using one of the methods from computational distributional semantics. As described below in more detail, we simulated semantic vectors. Thus, words are points in a k-dimensional semantic space, and the k 'axes' of this space function as the semantic predictors of the words.

From $\boldsymbol{S}$ and $\boldsymbol{C}$, we derive two hat matrices, one for comprehension ($\boldsymbol{H}_c$) and one for production ($\boldsymbol{H}_p$):

(11)
$$\boldsymbol{H}_c = \boldsymbol{C}(\boldsymbol{C}^{T}\boldsymbol{C})^{-1}\boldsymbol{C}^{T},$$

(12) $H_p = S(S^T S)^{-1} S^T.$

These hat matrices in turn allow us to estimate the semantic vectors predicted by the form vectors $\hat{S}$ and the form vectors predicted by the semantic vectors $\hat{C}$:

(13) $\hat{S} = H_c S,$

(14) $\hat{C} = H_p C.$

The R code for these calculations is:

```
Hcomp = C %*% ginv(t(C)%*%C)%*%t(C)
Hprod = S %*% ginv(t(S)%*%S)%*%t(S)
Shat = Hcomp %*% S
Chat = Hprod %*% C
```

where we use the generalized inverse to invert matrices. For the present simple example, the predicted matrices are identical to the 'observed' matrices.

4. Matrices for form and meaning in Latin

To implement the linear mappings between form and meaning in Latin, we first load the Latin dataset in the R package **WpmWithLdl.**

```
library(WpmWithLdl)
data(latin)
head(latin)
```

```
      Word Lexeme Person Number   Tense  Voice Mood
1    vocoo vocare     p1     sg present active  ind
2   vocaas vocare     p2     sg present active  ind
3    vocat vocare     p3     sg present active  ind
4 vocaamus vocare     p1     pl present active  ind
5 vocaatis vocare     p2     pl present active  ind
6   vocant vocare     p3     pl present active  ind
```

In this dataset, inflected word forms are listed in the first column **Word.** The second column **Lexeme** specifies the verb (indexed by the infinitive), while the remaining columns specify words' inflectional features. Before we can study mappings between form and meaning, and between meaning and form, we have to define the matrices representing words' forms and meanings. The pertinent matrices, C and S, that we examine in this study are calculated with the function **learn_mappings.** This function requires a formula as its first argument, in which all the classes (i.e., column names of the content lexomes and inflectional features) required to construct the semantic matrices should be included. (Note that the column name of the content lexome should always be the first element in the formula.) The directive `grams = 3` requests triphones as sublexical units of form.

```
m = learn_mappings(~ Lexeme + Person + Number + Tense + Voice + Mood,
                   data = latin,
                   grams = 3)
```

The object `m` is a list with a series of data objects relevant for modeling with LDL. The $\boldsymbol{C}$ and $\boldsymbol{S}$ matrices are included in this list. For convenience, we extract the matrices from `m`.

```
C = m$C
S = m$S
```

Details on how exactly the form matrix is constructed from the dataset can be found in the code and documentation of the `make_C_and_T_matrices` function. The upper left corner of the $\boldsymbol{C}$ matrix

```
C[1:8, 1:10]
```

```
       #vo voc oco coo oo# oca caa aas as# cat
vocoo    1   1   1   1   1   0   0   0   0   0
vocaas   1   1   0   0   0   1   1   1   1   0
vocat    1   1   0   0   0   1   0   0   0   1
vocaamus 1   1   0   0   0   1   1   0   0   0
vocaatis 1   1   0   0   0   1   1   0   0   0
vocant   1   1   0   0   0   1   0   0   0   0
clamoo   0   0   0   0   1   0   0   0   0   0
clamaas  0   0   0   0   0   0   0   1   1   0
```

illustrates how the form of word i is specified as a binary vector with a 1 in cell c_{ij} coding the presence of the j-th triphone in that word. The matrix with words' semantic vectors has the same layout,

```
S[1:8, 1:6]
```

```
                S1        S2         S3       S4        S5         S6
vocoo     1.592302  4.922176 -17.843929 24.26317 -7.730605  8.4711810
vocaas    6.352683  4.876391 -11.263886 29.82994 -8.820140  7.0834959
vocat    -1.848716  3.010773  -8.257137 26.58054 -4.947933  3.7924824
vocaamus  3.308614 10.414789 -25.433244 20.01480 -6.083262  1.1607091
vocaatis  7.309295 10.449995 -20.196866 26.59721 -8.302275 -0.8591813
vocant   -1.456805  7.218377 -17.234230 25.80952 -4.285191 -1.8888300
clamoo   -6.233772 -3.664891 -18.952648 12.45618 -6.196328  7.2665484
clamaas  -3.245372 -4.525140 -14.427739 19.93702 -6.849744  4.0173962
```

but the column features are no longer straightforwardly interpretable. For semantic features obtained with Latent Semantic Analysis (Landauer and Dumais, 1997), the features S_1, S_2, ... would represent latent variables. For the vector space used by Baayen et al. (2018), the features would reflect the association strength between the words in the rows of the matrix with those words in a corpus that show the highest variability in lexical co-occurrence. The present study makes use of simulated semantic vectors. One important reason for working with simulated semantic vectors for this study is that it is rarely the case that all possible inflected variants of a word are found in a corpus (see Karlsson, 1986; Loo et al., 2018a, b, for detailed discussion), and that as a consequence corpus data would be too sparse to derive proper semantic vectors.

One way of creating simulated vectors is to simply create a matrix with random numbers, without any correlational structure for the rows of the matrix. Such a matrix, however, does not do justice to the semantic similarities between words. For instance, one would expect inflectional variants of the same verb to be more similar to each other semantically than inflectional variants sampled from different verbs. To remedy this problem, in the `semantic_space` function, we assigned each lexome (including both content lexomes and all inflectional features) a semantic vector using standard normal random variates. The semantic vector of a given inflectional form was then the sum of its original random vector and the semantic vectors of all the pertinent lexomes. In this way, forms that share more lexomes would be more similar to each other than forms share less or no lexomes. The similarity structure among word forms can be seen in Figure 3, a heatmap of the correlation matrix for the row vectors of ***S***. In `R`, this heatmap is obtained with the following code, where we transpose ***S*** as in `R` correlations are calculated for column vectors.

```
heatmap(cor(t(S)))
```

The yellow blocks on the diagonal bring together forms with the same lexomes. For example, in the right-upper corner sit all plural word forms, and within that block are small clusters of word forms with the same person (p1, p2, p3). The details of this simulated matrix are not of interest. What is important is that the matrix reflects, in a high-dimensional vector space, the major morpho-syntactic similarities specified in the dataset. (Another possible solution to the lack-of-semantic-similarity problem in ***S*** is to use the `mvrnorm` function from the **MASS** package. This function takes as one of its inputs a variance-covariance matrix. By choosing appropriate covariances in the variance-covariance matrix, we can generate row vectors of ***S*** with a correlational structure that respects the morpho-syntactic similarity of the word forms. Incorporating this method is possible in the `semantic_space` function by setting `with_mvrnorm = TRUE`. More details can be found in the documentation of this function.)

5. Mappings between form and meaning

We are now ready to consider mappings from ***C*** (form) to ***S*** (meaning), and back from ***S*** to ***C***. Note that these mappings will almost always be different, compare,

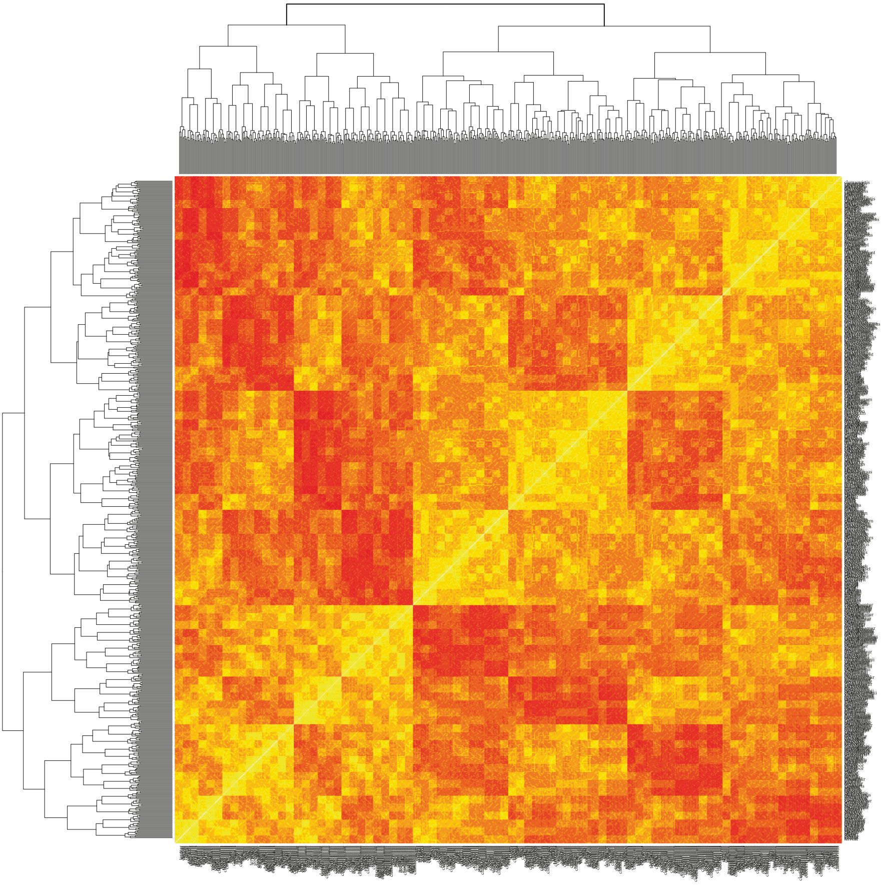

Figure 3. Heatmap for the correlation matrix of the row vectors of the simulated semantic matrix $\boldsymbol{S}$. The supplementary materials include a pdf of this heatmap with a larger granularity such that the individual forms can be read off on the bottom and right side of the plot. Brighter yellow indicates stronger positive correlations

e.g., the mappings $\boldsymbol{F}$ and $\boldsymbol{G}$ in Figure 2. And although in this simple example, $\boldsymbol{F}$ and $\boldsymbol{G}$ are each other's exact inverse, for more realistic and large matrices, this will no longer be the case, even though the inverse of, e.g., $\boldsymbol{G}$ may similar to a considerable extent to $\boldsymbol{F}$.

5.1 Comprehension: From form to meaning

Using Equation (11), we obtain the predicted semantic vectors $\hat{\boldsymbol{S}}$ by first calculating the hat matrix $\boldsymbol{H}$ and then multiplying this matrix with the semantic matrix $\boldsymbol{S}$.

```
Hcomp = C %*% ginv(t(C) %*% C) %*% t(C)
Shat = Hcomp %*% S
```

For evaluating model performance for word *i*, we compare a given predicted vector $\hat{s}_i$ with the semantic vectors s_j of all words $j = 1, 2, \ldots, n$, using the Pearson correlation. The word for which this correlation is greatest is selected as the word meaning that is recognized. The function `accuracy_comprehension` carries out these comparisons, and returns a list with both accuracy across all word forms and the correlation coefficients of the selected predicted vectors $\hat{s}_s$ with the targeted vectors s_i. Performance turns out to be 100% accurate.

```
res = accuracy_comprehension(m)
res$acc
```

```
[1] 1
```

There is a small number of homophones in the Latin dataset:

```
forms=table(latin$Word)
forms[forms>1]
```

```
audiam audiar carpis curris sapiam sapiar
     2      2      2      2      2      2
```

For these homophones, `accuracy_comprehension` selects the paradigm form with the best supported features as the word recognized. For *audiam,* for instance, we have the following ranking of forms:

```
semantic_neighbors("audiam")
```

```
           form         r        Word Lexeme Person Number      Tense
361      audiam 0.9451450      audiam audire     p1     sg    present
49       audioo 0.9113651      audioo audire     p1     sg    present
265      audiam 0.8622304      audiam audire     p1     sg     future
553   audiiverim 0.8328309  audiiverim audire     p1     sg    perfect
457    audiirem 0.8266663    audiirem audire     p1     sg       past
649 audiivissem 0.8245681 audiivissem audire     p1     sg pluperfect
61     audieebam 0.8029150   audieebam audire     p1     sg       past
363      audiat 0.7949122      audiat audire     p3     sg    present
    Voice Mood
361 active subj
49  active  ind
265 active  ind
553 active subj
457 active subj
649 active subj
61  active  ind
363 active subj
```

In the above output, *r* refers to the correlation of the predicted semantic vector with the observed semantic vector of the paradigm form. As long as the two homophones are assumed to be truly identical in form (see Gahl, 2008; Plag et al., 2017, for discussion of why this assumption may need to be relaxed) they will map onto exactly the same semantic vector. Thus, for any pair of homophones, one paradigm form will be mapped on the semantic vector for the same paradigm slot, whereas the other form will miss out on its own semantic vector and will instead be confused with the paradigm slot of its dominant homophonic twin. Without contextual disambiguation, there is no way in which the model could have performed better.

The use of semantic vectors might seem overly cumbersome given that one could set up, for each word, an indicator matrix of lexical and inflectional features. Such a matrix can be extracted from the `m` object:

```
T = m$ms3$matrices$T
dim(T)
```

```
[1] 672 22
```

```
T[1:8, 1:10]
```

```
         vocare clamare terrere docere audire sapire carpere currere p1 p2
vocoo         1       0       0      0      0      0       0       0  1  0
vocaas        1       0       0      0      0      0       0       0  0  1
vocat         1       0       0      0      0      0       0       0  0  0
vocaamus      1       0       0      0      0      0       0       0  1  0
vocaatis      1       0       0      0      0      0       0       0  0  1
vocant        1       0       0      0      0      0       0       0  0  0
clamoo        0       1       0      0      0      0       0       0  1  0
clamaas       0       1       0      0      0      0       0       0  0  1
```

This matrix has far fewer columns than ***S***. A mapping from ***C*** to ***T*** is possible

```
That = Hcomp %*% T
```

and is succesful:

```
n_identical_features = rep(0, nrow(latin))
for (i in 1:nrow(latin)) {
  n_identical_features[i] = sum(round(That[i,]) == T[i,])
}
table(n_identical_features)
```

```
n_identical_features
 18  20  21  22
  4   1   2 665
```

The only forms for which the observed and predicted features are not identical are, unsurprisingly, the six homophones.

```
latin[n_identical_features != 22,]
```

```
      Word  Lexeme Person Number   Tense   Voice Mood
```

```
74  carpis carpere     p2     sg present  active  ind
265 audiam  audire     p1     sg  future  active  ind
271 sapiam  sapire     p1     sg  future  active  ind
313 audiar  audire     p1     sg  future passive  ind
319 sapiar  sapire     p1     sg  future passive  ind
374 carpis carpere     p2     sg present  active subj
380 curris currere     p2     sg present  active subj
```

In five out of the six homophones, the features of the dominant homophonic twin are selected. The only exception is *carpis,* where the `Mood` features of the two homophonic words (`ind` and `subj`) are equally supported for both words, suggesting that both senses of this word form are equally activated.

```
round(That[which(latin$Word=="carpis"),],2)
```

```
       vocare clamare terrere docere audire sapire carpere currere p1 p2
carpis      0       0       0      0      0      0       1       0  0  1
carpis      0       0       0      0      0      0       1       0  0  1
       p3 sg pl present past perfect future pluperfect active passive ind
carpis  0  1  0       1    0       0      0          0      1       0 0.5
carpis  0  1  0       1    0       0      0          0      1       0 0.5
       subj
carpis  0.5
carpis  0.5
```

Whereas it is possible to map from ***C*** to ***T***, an accurate mapping in the reverse direction is not possible, as we shall see below. The problem that we encounter here is that a linear mapping from a lower-dimensional space into a higher-dimensional space can only have an image in that larger space that is of the same dimensionality (rank) as the lower-dimensional space (Kaye and Wilson, 1998). For instance, a linear mapping from a space $\{A\}$ in $\mathbb{R}^2$ to a space $\{B\}$ in $\mathbb{R}^3$ will result in a plane in $\{B\}$. All the points in $\{B\}$ that are not on this plane cannot be reached from $\{A\}$.

A further problem associated with predicting lexical and inflectional features is that, as mentioned above, the matrix ***T*** provides a very impoverished representation of the semantics of the Latin inflectional forms. In Word and Paradigm Morphology, morpho-syntactic features do not represent semantic primitives, but rather distribution classes. Such distributional classes are much better represented by semantic vectors than by binary coding for inflectional features. The excellent performance observed for the mapping of ***C*** onto ***S***, rather than the mapping from ***C*** to ***T*** , therefore, is the result of primary interest.

5.2 Production: from meaning to form

For speech production, we consider the mapping from the semantic matrix ***S*** onto the form vectors of ***C*** . The pertinent hat matrix ***H*** is

```
Hprod = S %*% ginv(t(S) %*% S) %*% t(S)
```

With $\boldsymbol{H}$, we obtain the predicted form vectors $\hat{\boldsymbol{C}}$

```
Chat = Hprod %*% C
rownames(Chat)=rownames(C)
round(Chat[1:8, 1:8],2)
```

```
           #vo   voc   oco   coo  oo#  oca  caa   aas
vocoo     1.00  1.00  0.44  0.42 0.50 0.44 0.49 -0.02
vocaas    1.01  1.01  0.06  0.04 0.04 0.91 0.90  0.43
vocat     1.00  1.00  0.00  0.02 0.02 0.97 0.50 -0.02
vocaamus  1.00  1.00  0.03  0.03 0.06 0.84 0.79 -0.10
vocaatis  1.02  1.02 -0.01  0.01 0.01 1.01 0.96  0.16
vocant    1.00  1.00 -0.03 -0.03 0.03 0.87 0.43  0.00
clamoo   -0.01 -0.01  0.00  0.00 0.46 0.15 0.20  0.09
clamaas   0.00  0.00 -0.02 -0.03 0.13 0.02 0.07  0.40
```

The resulting form vectors provide information on the amount of support for the individual triphones, but do not, as such, indicate how the triphones should be ordered to obtain a proper characterization of a word's form.

We therefore need to consider all the ways in which phone triplets can be joined into legal word forms. Triphones contain intrinsic order information: a triphone such as `abc` can be joined with `bcd` but not with `dcb`. We can exploit this partial ordering information efficiently using graph theory.

We assign triphones to the vertices of a graph, and connect these vertices with a directed edge when the corresponding triphones have the proper overlap (`bc` for `abc` and `bcd`). Figure 4 shows the graph of all the triphones contained in the Latin dataset, and the triphone path of the word *sapiivisseemus* is marked in red. The path starts with the word-inital triphone `#sa` and ends with the word-final triphone `us#` (with the `#` symbol representing word boundary). Each edge in the graph is associated with a weight. For a given word *i*, these weights are taken from the predicted form vector $\hat{\boldsymbol{c}}_i$ (the row vector of $\hat{\boldsymbol{C}}$ corresponding to the semantic vector $\boldsymbol{s}_i$ that is the input for production). Using *j* and *k* to index the positions of triphones in the columns of $\hat{\boldsymbol{C}}$, the weight on an edge from triphone t_j to triphone t_k is set to $\hat{c}_{ik}$, i.e., to the *k*-th value in the predicted form vector $\hat{c}_i$. The support for a path in the graph can now be defined as the sum of the weights on the edges of this path. Importantly, from a word's predicted form vector $\hat{\boldsymbol{c}}_i$, we calculate all *m* paths $p_1, p_2, \ldots, p_{\mathrm{m}}$ $(m \geq 1)$ with path weights $\omega_1, \omega_2, \ldots, \omega_m$ that start with an initial triphone and end with a final triphone.

To find these paths, we make use of the **igraph** package (Csardi and Nepusz, 2006), which provides the `all_simple_paths` function to trace all paths that start from a given vertex and that do not contain cycles. (As cycles can be traversed ad libitum, paths with cycles cannot be enumerated.) From all simple paths, we select those that proceed from an initial triphone to a final triphone. A word's

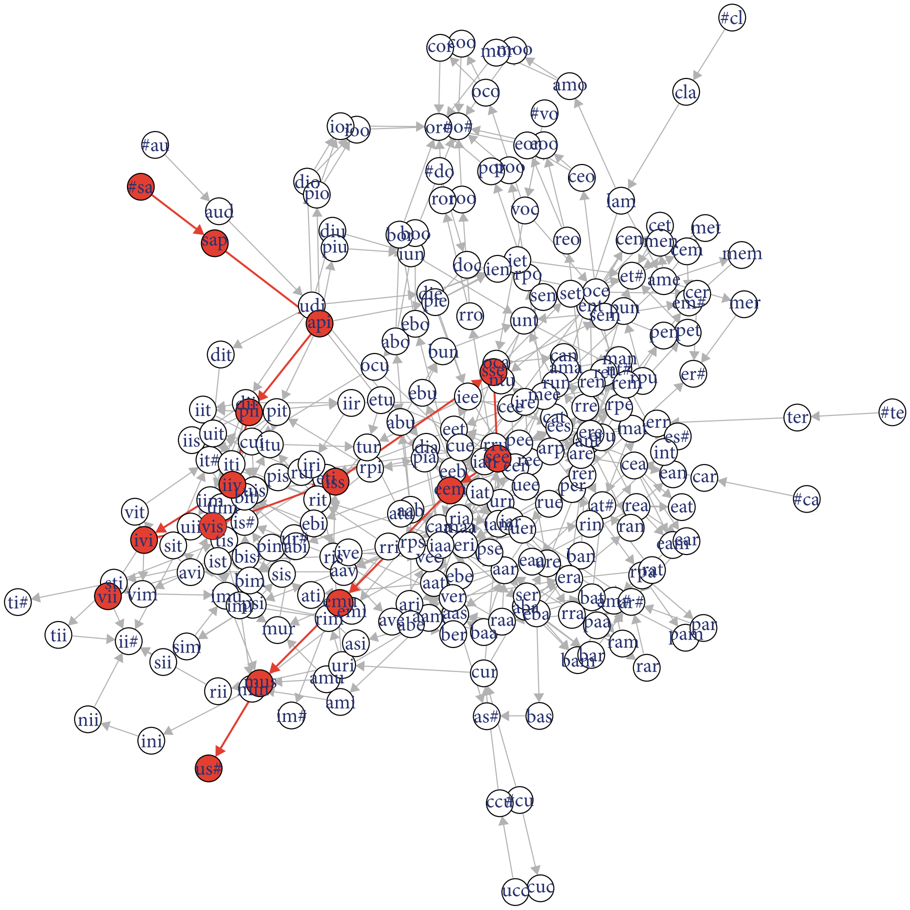

Figure 4. The triphone graph of *sapiivisseemus*. The path that links all the triphones in the word is highlighted in red

path can contain a cycle, however, as illustrated in Figure 5 for *terreerees*. (In this figure, for representational clarity, vertices and edges that have little or no support from the mapping from meaning to form have been suppressed.) Cycles of length two and length three can be extracted from the graph with the functions `which_mutual` and `triangles` from the **igraph** package. The function `speak` from **WpmWithLdl** inserts such cycles, if present, into any path where this is possible. Any new (and longer) paths found are added to the list of paths, and their associated path weights are calculated. It turns out that in order to avoid paths with unjustified repeated cycles, a path with a cycle is accepted as valid only if there is one, and only one, vertex of that cycle that is encountered twice in that path.

At this point, we have a set of candidate paths, each of which corresponds to a word that can be selected for articulation. The question is how to select the

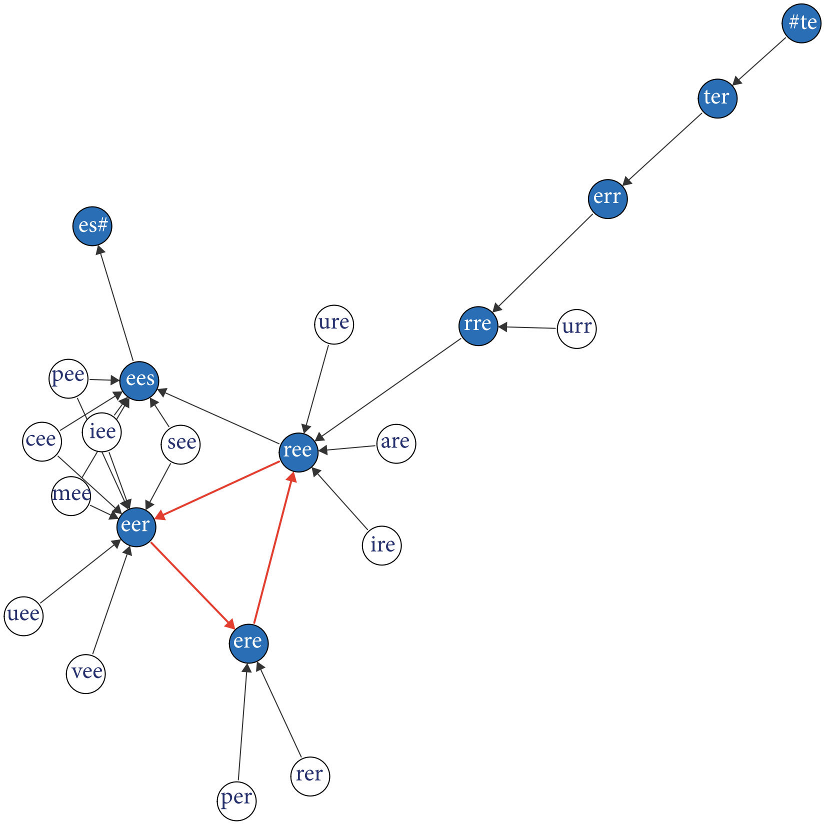

Figure 5. The triphone graph of *terreerees*. The triangle (3-step cycle) in the path is highlighted in red

proper word. The weight of a path is not a good guide for selection, as longer paths typically have high path weights. A heuristic that works quite well, but not without error, is to adjust path weights for path length. What works even better is 'synthesis by analysis'. For each candidate path, we take the triphones and map these back onto semantic vectors using the comprehension model. Each of the resulting semantic vectors can then be correlated with the semantic vector that is targeted for realization in speech. The path that generates the highest correlation can now be selected for articulation. For a general framework within which this interplay of comprehension and production receives justification, see Hickok (2014).

A snag that has to be dealt with at this point is what to do when the highest correlation is shared between multiple candidate paths. Here, the edge weights come into play. Lower edge weights indicate weak links in the path, links that are not well supported by $\hat{C}$. We therefore want to avoid, as much as possible, paths with weak edge weights. We also want to avoid unnecessary long paths. Let R denote the ratio of a path's length divided by the smallest edge weight in the vector of edge weights $\boldsymbol{\omega}$ of that path:

$$R = \frac{\#\text{vertices}}{\min(w)},$$

henceforth the length to weakest link ratio. We select for production that path for which R is smallest.

This algorithm is implemented in the function `speak` of **WpmWithLdl.** We illustrate its use for the first word of the Latin dataset (`pos = 1`).

```
s1  = speak(pos = 1, grams = 3, threshold = 0.1,
            amat = m$am, data = latin,
            C = m$C, S = m$S,
            Bcomp = m$Bcomp, Chat=m$Chat)

s1$candidates

[1] "vocoo"      "vocaaveram" "vocaaverat"

s1$prediction

[1] "vocoo"

s1$cors

    vocoo vocaaveram vocaaverat
1.0000000  0.8258359  0.7184197

s1$length_weakest_link_ratio

   vocoo vocaaveram vocaaverat
11.90476   71.42857   62.50000
```

In the call to `speak`, the `threshold = 0.1` directive thins the graph by removing all vertices for which $\hat{c}_{ij} < 0.1$. Without thinning, the number of possible paths that are calculated by `all_simple_paths` can become very large – even for the present small dataset, the number of paths can run into the thousands. Discounting of vertices and corresponding edges that are highly unlikely to contribute serious contending path candidates speeds up calculations substantially.

The `amat = m$am` directive tells `speak` where it can find the adjacency matrix that defines the graph. The adjacency matrix is calculated by `learn_mappings` and returned in its output list with under the name `am`. This output list also makes available the transformation matrix $\hat{\boldsymbol{B}}_c$, which is used to calculate the semantic vectors corresponding to the candidate paths.

To speed up evaluation for all forms in the Latin paradigms, we run `speak` in parallel on four cores.

```
library(doParallel)

no_cores=4
cl <- makeCluster(no_cores)
registerDoParallel(cl)
rs = foreach(i=1:nrow(latin), .packages=c("igraph", "WpmWithLdl")) %dopar%
   speak(pos = i, grams = 3, threshold = 0.1,
         amat = m$am, data = latin,
         C = m$C, S = m$S,
         Bcomp = m$Bcomp, Chat=m$Chat)
stopCluster(cl)
```

Model accuracy is straightforwardly evaluated by comparing the predicted words with the observed words.

```
preds = sapply(rs, FUN = function(lis) lis$prediction[1])
dfr = data.frame(preds = unlist(preds),
                 obs = latin$Word,
                 stringsAsFactors=FALSE)
sum(dfr$preds == dfr$obs)

[1] 670
```

For 670 out of 672 paradigm cells, the predicted form corresponds to the targeted observed form. The two cases where the model misses the correct form are

```
dfr[dfr$preds!=dfr$obs,]

          preds         obs
428  curriaaris   curraaris
524 curreereris currereeris
```

The first of these, *curriaaris,* is a non-existent form that, remarkably, perfectly predicts the semantic vector of its paradigm slot.

```
rs[[which(latin$Word=="curraaris")]]$cors

   curreetur      curris   curriaaris curriaareetur    curraaris
   0.5891857   0.6920681    0.9915895    0.6378089      0.9248290
curraareetur
   0.5774105
```

The attractor for *curriaaris* is *audiaaris* (a form from the fourth instead of the third conjugation):

```
latin[grep("iaaris", latin$Word),]

         Word Lexeme Person Number   Tense   Voice Mood
410 audiaaris audire     p2     sg present passive subj
```

In the case of *curreereris,*

```
head(round(rs[[which(latin$Word=="currereeris")]]$cors, 2),8)

  curreebaaminii      curreeris       curreereris     curreris
            0.67           0.66              1.00         0.66
currereebaaminii    currereeris cuccurreebaaminii cuccurreeris
            0.79           1.00              0.55         0.57

round(rs[[which(latin$Word=="currereeris")]]$length_weakest_link_ratio, 2)[c(3,6)]
currereeris currereeris
      36.67       36.67
```

we turn out to have a perfect tie, with a non-existing form pre-empting (by alphabetic order) the correct form. This tie can be resolved by replacing triphones by 4-phones, in which case production performance is completely error-free.

What does not work at all is mapping from the ***T*** matrix specifying morphosyntactic features to the ***C*** matrix with triphones (or 4-phones).

```
Hprod = T %*% ginv(t(T)%*%T)%*%t(T)
Chat = Hprod %*% C
n_identical_features = rep(0, nrow(latin))
for (i in 1:nrow(latin)) {
  n_identical_features[i] = sum(round(Chat[i,]) == C[i,])
}
table(n_identical_features)


n_identical_features
244 245 246 247 248 249 250 251 252 253 254
  2   6  17  32  53  69 117 151 136  70  19
```

There are only 19 forms for which all 254 features are correctly supported. It is impossible to map, with any reasonable degree of accuracy, the low-dimensional space {T} onto the high-dimensional space {*C*} when, as in the present study, the mapping is constrained to be linear.

6. A novel perspective on traditional evidence for morphemes

We have shown that, once meanings and forms are reconceptualized as points in spaces with high dimension k in $\mathbb{R}^k$, it is possible to set up linear mappings from form to meaning, and from meaning to form. These simple linear mappings achieve an astonishingly high degree of accuracy, without having to define stems, morphemes or exponents, theme vowels, and inflectional classes.

This result raises the question of how to interpret the large body of experimental results that has been argued to support the psychological reality of the morpheme. To address this question, we begin with noting that the experimental literature typically reasons as follows.

First, a hypothesis is set up which has a conditional form: "if morphemes are real (P), we should observe effect Q". For instance, if words consist of morphemes as beads on a string, we should expect to find some kind of probabilistic gaps between the morphemes in a word. Such gaps can be operationalized through low transitional probabilities at morpheme boundaries (Hay, 2002; Hay and Baayen, 2003). Now some experiment is run, and effect Q is indeed observed. For instance, at morpheme boundaries, inter-keystroke intervals are longer (Weingarten et al., 2004; Bertram et al., 2015). Having observed effect Q, the conclusion drawn is that morphemes must indeed exist (P).

However, two premises are at issue here. The first is the conditional claim that if morphemes exist, effect Q should follow ($P \rightarrow Q$). The second premise is that morphemes exist (P). When both premises are true, it follows that effect Q must exist:

$$P \rightarrow Q, P \; ? \; Q.$$

However, from observing effect Q, and given the validity of premise $P \rightarrow Q$, we cannot conclude that morphemes exist (P). Affirming the consequent

$$P \rightarrow Q, Q \; ? \; P$$

is a fallacious line of reasoning, as P was never asserted as the only possible condition for Q. To continue with the current example of morpheme boundary effects, Baayen et al. (2018) show that such effects at morpheme junctions can arise in the present theory, even though this theory eschews morphemes or exponents alltogether (see also Seidenberg, 1987). The reason is that in a word's graph, edge weights tend to be substantially lower when at the end of a stem paths fork to support different inflectional variants. The lower the edge weights at such forks are, the more costly the production of the triphone that the edge connects to.

One kind of evidence that has been advanced to support morphemes comes from neuroimaging studies. For instance, Bozic et al. (2010) compared the Bold response in fMRI to simple English words (e.g., *cream)* on the one hand with the Bold response to words with a final *d* or *s*, which typically realizes number and tense inflection in English (e.g., *played, packs).* They reported that (potentially) inflected words gave rise to a stronger Bold response, compared to their control condition, in the left inferior frontal area BA 45, and conclude from this that

> that left frontal regions perform decompositional computations on grammatical morphemes, required for processing inflectionally complex words.
>
> (Bozic and Marslen-Wilson, 2010, p. 5–6)

The logic of their reasoning is that if decompositional morphological processes exist, there must be brain areas that are differentially engaged. Having found differentiation in BA 45, the conclusion is that this area performs morphological decomposition. However, the conclusion that morphological decomposition exists (and apparently, for whatever reason, is executed in BA 45) is premature, as it is possible that area BA 45 lights up in an fMRI scanner for reasons other than morphological decomposition.

Given that there are good reasons to reject the morpheme as a useful theoretical construct, and given that even exponents may not be required, we have to address the question of what factors other than morphological decomposition could give rise to topologically concentrated Bold responses in fMRI scans. It is at present unclear whether the fMRI effects reported are actually trustworthy, given the replicability crisis in both psychology and neuroimaging (Button et al., 2013; Open Science Collaboration, 2015) and the laxness of the criteria in neuroimaging with respect to by-item and by-subject variability. Nevertheless, the issue of the topological organization of linguistic representations is of theoretical interest, and has been pursued for morphology by Pirelli and collaborators using temporal self-organizing maps (TSOMs Ferro et al., 2011; Chersi et al., 2014; Pirrelli et al., 2015). In TSOMs, word forms become paths in a 2-D plane. However, in our experience, self-organizing maps do not scale up well to realistically sized lexicons.

Within the present framework, the question of the topological organization of morphological form can be addressed straightforwardly at the level of triphones. As mentioned above, we opted for triphones as central form features for two reasons. First, triphones do justice to the modulation of speech sounds by their context. The place of articulation of stops is retrievable from formant transitions present in adjacent vowels. Second, triphones provide rich information about sequencing, which we have exploited to construct wordform paths in triphone graphs. Thus far, however, we have remained agnostic about the topological organization of the vertices in these graphs. If triphones have some cognitive reality, and if there are cell assemblies subserving triphone-like units, then it makes sense to reflect on the spatial organization of triphones on a surface. In what follows, we make the simplifying assumption, just as TSOMs do, that this surface is a plane in $\mathbb{R}^2$.

To obtain a mapping of triphones onto a plane in $\mathbb{R}^2$, we make use of an algorithm from physics that has been transferred to graph theory with the goal of obtaining well-interpretable layouts of large graphs, graphopt (http://www.schmuhl.org/graphopt/). The graphopt algorithm, which has also been implemented in the **igraph** package, uses basic principles of physics to iteratively determine an optimal layout. Each node in the graph is given both mass and an electric charge, and edges between nodes are modeled as springs. This sets up a system in which there are attracting and repelling forces between the vertices of the graph. This physical system is simulated until it reaches equilibrium. In other words, the graphopt algorithm provides us with a simple way of implementing spatial self-organization of triphones.

Figure 6 (produced with the function `graph_topology_flect`) presents the results of the graph-opt algorithm applied to the triphones of the Latin dataset. Triphones that are unique for a given inflectional function (PERFECT, PLUPERFECT, FUTURE, and PAST) are highlighted. Interestingly, triphones show considerable topological clustering depending on what inflectional function they subserve. For instance, triphones unique for the PERFECT are strongly represented in the upper part of the graph. This clustering, however, does not imply that it

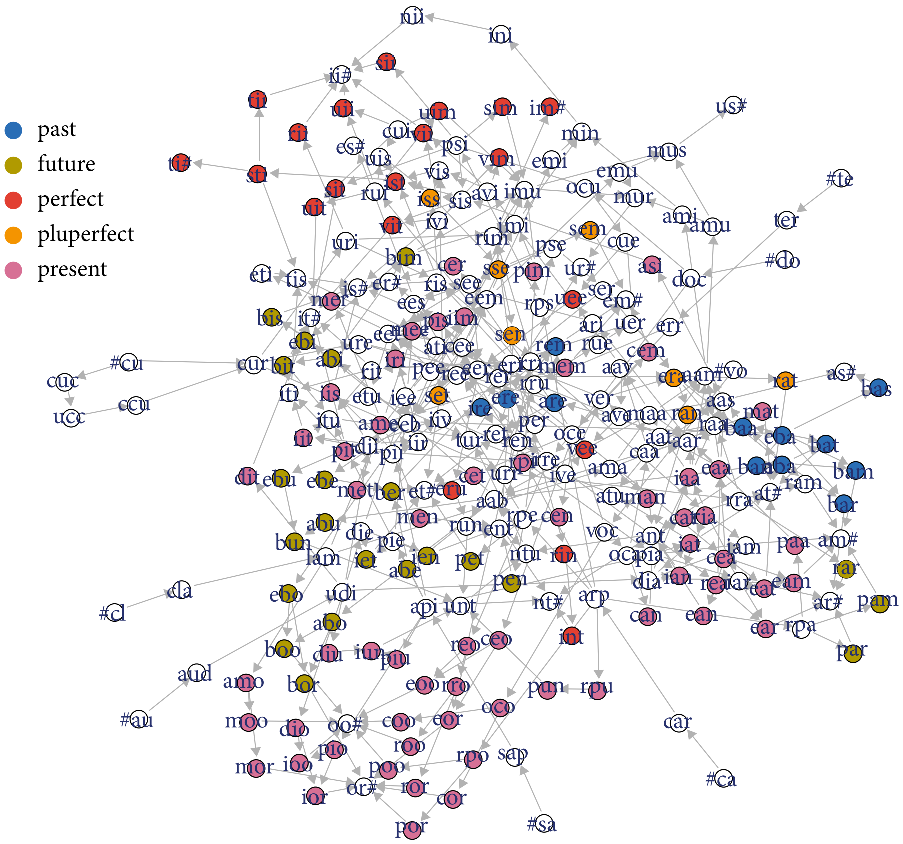

Figure 6. Topology of tenses in the triphone graph of the Latin dataset, using the `graphopt` layout. Note the local topological coherence of the triphones subserving the perfect, the pluperfect, the future, and the past

is in the upper part of the plane that inflected forms realizing the perfect are decomposed or combined. The clustering arises as a consequence of physical constraints on low-level units that have to be allocated to a 2-D plane. Figure 7 (obtained with the `graph_topology_segments` function) shows that in the very same network, triphones that are unique to the five vowels of Latin also show considerable clustering. (For phoneme-like clustering in the cortex, see Cibelli et al. (2015) and references cited there.) One might take clusterings like this to imply that phonemes exist in our model, yet there are no units for phonemes at all in the system that underlies the graph, but only sequences of triples of phones – and these units have a very different theoretical motivation than the phoneme in structuralist linguistics and offshoots thereof. A conjecture that follows from the present results is that the details of how triphones self-organize is highly dependent on how a language organizes its morphology.

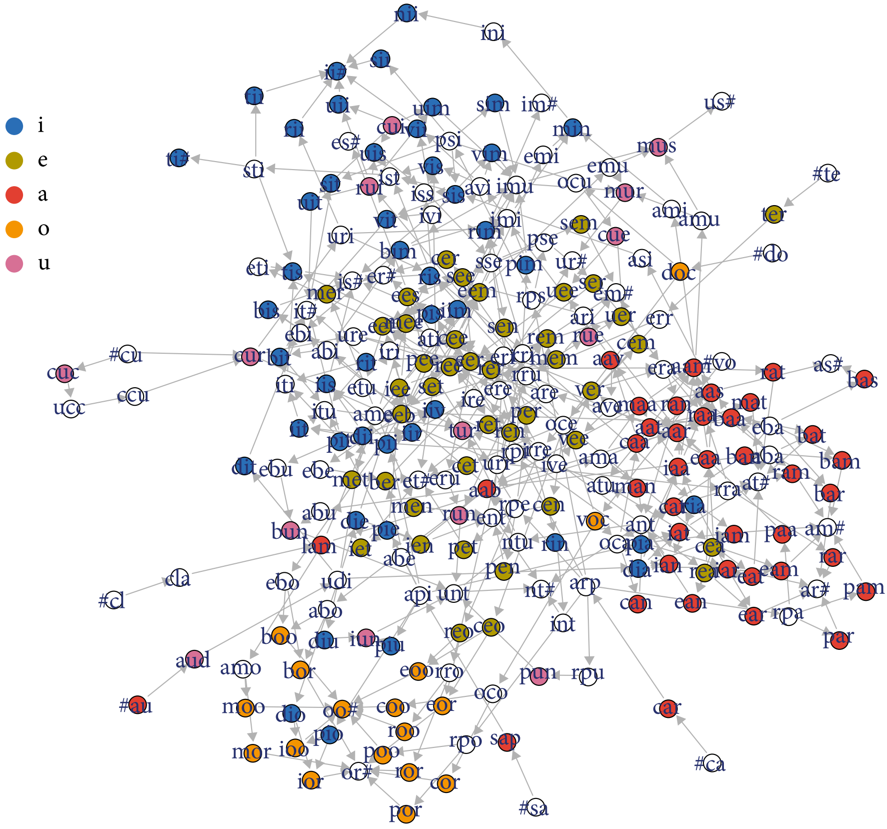

Figure 7. Topology of vowels in the triphone graph of the Latin dataset, using the same `graphopt` layout as for Figure 6

We end this section with a brief note on double dissociations in aphasia and their implications for understanding the nature of morphological processing. Selective impairments, such as relatively unimpaired performance on irregular forms and impaired performance on regulars, are often taken as providing evidence for distinct processing components. However, as shown by Juola (2000), double dissociations can arise in non-modular neural networks when across many simulation runs the network is lesioned randomly. We expect that similar results are within reach for the present approach, as linear mappings are equivalent to two-layer linear networks. Furthermore, when topological organization is imposed, as in Figure 6, local lesions in for instance the upper left of this figure, can have an effect that is easily misunderstood as the perfect tense morpheme (or exponent) having been lost.

7. Concluding remarks

We have shown that it is computationally possible to map 672 different inflectional forms of Latin onto their semantic vectors, and to map these semantic vectors back onto their corresponding word forms, without requiring constructs such as stems, affixes, exponents, morphemes, allomorphs, theme vowels, and inflectional classes. We have also shown that when the basic units of analysis, triphones, are allowed to self-organize in a 2-dimensional plain, patterns emerge that are reminiscent of morphemes (or exponents) and phonemes, without such units being part of the generating system.

Interestingly, it is apparently not necessary to harness the power of deep learning to see clusterings resembling traditional units emerge. Together with the results of a much larger-scale study for English (Baayen et al., 2018), we conclude it is worthwhile to explore in further detail the potential of linear mappings for our understanding of lexical processing in comprehension and speech production.

Some caveats are in order, however. In actual language use, it is typically the case that only a minority of the set of possible inflectional forms is used, as shown by Karlsson (1986) and Loo et al. (2018a, b). For more realistic modeling of Latin inflectional morphology, a corpus of Latin will be indispensible. A corpus will not only allow us to test how well the present theory works for sparse data, but it will also make it possible to start working with actual semantic vectors rather than simulated ones. Given a corpus-informed model, it will also become possible to address the productivity of the system (see Baayen et al., 2018, for more detailed discussion and computational implementation) and the role of principal parts.

Undoubtedly, there are limits to what can be done with simple linear mappings. Possibly, these limits constrain morphological form. Even in Latin, periphrastic constructions were in use, possibly in order not to stretch the fusional system beyonds its limits. It is also conceivable that there are black holes in a system, areas where uncertainties about which path to follow are too great to allow production with confidence. A possible case in point is the Polish genitive, see Dabrowska (2001) for detailed discussion of the problem and Divjak and Milin (2018) for a possible solution.

A further complication is that human performance is not error-free. The larger context within which words are used is of vital importance for both comprehension and production. One of the reasons that words 'by themselves' are fragile is the high variability of word forms, a factor that we have ignored alltogether in this study. This variability characterizes not only the spoken word (Johnson, 2004), but also printed words. A printed word (given font and font size) might seem invariable, but how it actually appears on the retina depends on angle and viewing distance. As a consequence, there is much more variability in actual lexical processing than in the present simulations, and as a consequence the present practically error-free performance observed for small simulated data sets does not carry over to full-scale lexical processing (for detailed discussion and modeling results for auditory comprehension of real speech, see Arnold et al. (2017); Baayen et al. (2018); Shafaei Bajestan and Baayen (2018)).

Nevertheless, the results obtained thus far are promising in that they suggest that linear mappings between vectors of form, anchored in triphones, and vectors of reals representing meaning are surprisingly effective. In this study, we have made use of simulated semantic vectors, constructed in such a way that vectors of words that share more features are more similar to each other. Interestingly, performance does not degrade for random semantic vectors, which is mathematically unsurprising as uncorrelated semantic vectors are less confusable and hence afford more accurate mappings. Importantly, once we drop the assumption that meanings are symbolic and representable by isolated units, or by highly specific semantic logical functions, and that instead we start entertaining the possibility that meanings are representations in a high-dimensional vector space, then mappings between form and meaning become much simpler. This result holds irrespective of whether meanings are equated with distributional vector space models constructed from corpora, or whether such vectors are seen as imperfect probes of much richer distributional models of our experience of the world. Thus, linear discriminative learning offers a new perspective on the functional architecture of lexical processing in comprehension and production.

Funding

This research was supported by an ERC advanced Grant (no. 742545) to the first author.

References

Ackerman, F., Blevins, J. P., and Malouf, R. (2009). Parts and wholes: Implicative patterns in inflectional paradigms. In Blevins, J. P. and Blevins, J., editors, *Analogy in grammar: Form and acquisition*, pages 54–82. Oxford Univ. Press Oxford, UK. https://doi.org/10.1093/acprof:oso/9780199547548.003.0003

Arnold, D., Tomaschek, F., Lopez, F., Sering, T., and Baayen, R. H. (2017). Words from spontaneous conversational speech can be recognized with human-like accuracy by an error-driven learning algorithm that discriminates between meanings straight from smart acoustic features, bypassing the phoneme as recognition unit. *PLOS ONE*, *12*(4):e0174623. https://doi.org/10.1371/journal.pone.0174623

Aronoff, M. and Fudeman, K. (2011). *What is morphology?*, volume *8*. John Wiley & Sons.

Baayen, R. H., Chuang, Y.-Y., Shafaei-Bajestan, E., and Blevins, J. (2018). The discriminative lexicon: A unified computational model for the lexicon and lexical processing in comprehension and production grounded not in (de)composition but in linear discriminative learning. *Complexity*, accepted for publication.

Beard, R. (1977). On the extent and nature of irregularity in the lexicon. *Lingua*, *42*:305–341. https://doi.org/10.1016/0024-3841(77)90102-4

Bertram, R., Tønnessen, F. E., Strömqvist, S., Hyönä, J., and Niemi, P. (2015). Cascaded processing in written compound word production. *Frontiers in human neuroscience*, *9*:207.

Blevins, J. P. (2003). Stems and paradigms. *Language*, *79*:737–767. https://doi.org/10.1353/lan.2003.0206

Blevins, J. P. (2006). Word-based morphology. *Journal of Linguistics*, *42*(03):531–573. https://doi.org/10.1017/S0022226706004191

Blevins, J. P. (2016). *Word and paradigm morphology*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199593545.001.0001

Booij, G. (2012). *The grammar of words: An introduction to linguistic morphology*. Oxford University Press.

Booij, G. E. (2016). Construction morphology. In Hippisley, A. and Stump, G., editors, *The Cambridge Handbook of Morphology*, pages 424–448. Cambridge University Press, Cambridge. https://doi.org/10.1017/9781139814720.016

Booij, G. (2018). *The Construction of Words: Advances in Construction Morphology*, volume *4*. Springer. https://doi.org/10.1007/978-3-319-74394-3

Bozic, M. and Marslen-Wilson, W. (2010). Neurocognitive contexts for morphological complexity: Dissociating inflection and derivation. *Language and Linguistics Compass*, *4*(11):1063–1073. https://doi.org/10.1111/j.1749-818X.2010.00254.x

Bozic, M., Tyler, L. K., Ives, D. T., Randall, B., and Marslen-Wilson, W. D. (2010). Bihemispheric foundations for human speech comprehension. *Proceedings of the National Academy of Sciences*, *107*(40):17439–17444. https://doi.org/10.1073/pnas.1000531107

Brown, D. and Hippisley, A. (2012). *Network morphology: A defaults-based theory of word structure*, volume *133*. Cambridge University Press. https://doi.org/10.1017/CBO9780511794346

Button, K. S., Ioannidis, J. P., Mokrysz, C., Nosek, B. A., Flint, J., Robinson, E. S., and Munafò, M. R. (2013). Power failure: Why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, *14*(5): 365–376. https://doi.org/10.1038/nrn3475

Butz, M.V. and Kutter, E.F. (2016). *How the mind comes into being: Introducing cognitive science from a functional and computational perspective*. Oxford University Press.

Cahill, L. and Gazdar, G. (1999). German noun inflection. *Journal of Linguistics*, *35*(1):1–42. https://doi.org/10.1017/S0022226798007294

Chersi, F., Ferro, M., Pezzulo, G., and Pirrelli, V. (2014). Topological self-organization and prediction learning support both action and lexical chains in the brain. *Topics in cognitive science*, *6*(3):476–491. https://doi.org/10.1111/tops.12094

Cibelli, E.S., Leonard, M.K., Johnson, K., and Chang, E.F. (2015). The influence of lexical statistics on temporal lobe cortical dynamics during spoken word listening. *Brain and language*, *147*:66–75. https://doi.org/10.1016/j.bandl.2015.05.005

Corbett, G.G. and Fraser, N.M. (1993). Network morphology: a datr account of russian nominal inflection. *Journal of linguistics*, *29*(1):113–142. https://doi.org/10.1017/S0022226700000074

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *Inter Journal*, Complex Systems:1695.

Dabrowska, E. (2001). Learning a morphological system without a default: the Polish genitive. *Journal of child language*, *28*:545–574. https://doi.org/10.1017/S0305000901004767

Divjak, D. and Milin, P. (2018). *Language, learning and usage-based linguistics*. Invited talk at the Philological Society, London, February 9, 2018.

Erelt, M., editor (2003). *Estonian language*. Estonian academy publishers, Tallinn.

Ernestus, M., Baayen, R.H., and Schreuder, R. (2002). The recognition of reduced word forms. *Brain and Language*, *81*:162–173. https://doi.org/10.1006/brln.2001.2514

Ferro, M., Marzi, C., and Pirrelli, V. (2011). A self-organizing model of word storage and processing: implications for morphology learning. *Lingue e linguaggio*, *10*(2):209–226.

Firth, J.R. (1968). *Selected papers of J R Firth, 1952–59*. Indiana University Press.

Gahl, S. (2008). Time and thyme are not homophones: The effect of lemma frequency on word durations in spontaneous speech. *Language*, *84*(3):474–496. https://doi.org/10.1353/lan.0.0035

Hay, J.B. (2002). From speech perception to morphology: Affix-ordering revisited. *Language*, *78*:527–555. https://doi.org/10.1353/lan.2002.0159

Hay, J.B. and Baayen, R.H. (2003). Phonotactics, parsing and productivity. *Italian Journal of Linguistics*, *1*:99–130.

Hickok, G. (2014). The architecture of speech production and the role of the phoneme in speech processing. *Language, Cognition and Neuroscience*, *29*(1):2–20. https://doi.org/10.1080/01690965.2013.834370

Hockett, C. (1960). The origin of speech. *Scientific American*, *203*:89–97. https://doi.org/10.1038/scientificamerican0960-88

Jackendoff, R. (1990). *Semantic Structures*. MIT Press, Cambridge.

Johnson, K. (2004). Massive reduction in conversational American English. In *Spontaneous speech: data and analysis. Proceedings of the 1st session of the 10th international symposium*, pages 29–54, Tokyo, Japan. The National International Institute for Japanese Language.

Juola, P. (2000). Double dissociations and neurophysiological expectations. *Brain and cognition*, *43*(1–3):257–262.

Karlsson, F. (1986). Frequency considerations in morphology. *STUF-Language Typology and Universals*, *39*(1–4):19–28. https://doi.org/10.1524/stuf.1986.39.14.19

Kaye, R. and Wilson, R. (1998). *Linear Algebra*. Oxford University Press.

Kemps, R., Ernestus, M., Schreuder, R., and Baayen, R. H. (2004). Processing reduced word forms: The suffix restoration effect. *Brain and Language*, *19*:117–127. https://doi.org/10.1016/S0093-934X(03)00425-5

Landauer, T. and Dumais, S. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, *104*(2):211–240. https://doi.org/10.1037/0033-295X.104.2.211

Lardier, D. (2014). Words and their parts. In Fasold, R. W. and Connor-Linton, J., editors, *An introduction to language and linguistics*. Cambridge University Press, Cambridge.

Levelt, W., Roelofs, A., and Meyer, A. S. (1999). A theory of lexical access in speech production. *Behavioral and Brain Sciences*, *22*:1–38. https://doi.org/10.1017/S0140525X99001776

Lieber, R. (1980). On the Organization of the Lexicon. PhD thesis, MIT, Cambridge.

Loo, K., Jaervikivi, J., and Baayen, R. (2018a). Whole-word frequency and inflectional paradigm size facilitate estonian case-inflected noun processing. *Cognition*, *175*:20–25. https://doi.org/10.1016/j.cognition.2018.02.002

Loo, K., Jaervikivi, J., Tomaschek, F., Tucker, B., and Baayen, R. (2018b). Production of estonian case-inflected nouns shows whole-word frequency and paradigmatic effects. *Morphology*, *1*(28):71–97. https://doi.org/10.1007/s11525-017-9318-7

Marantz, A. (2013). No escape from morphemes in morphological processing. *Language and Cognitive Processes*, *28*(7):905–916. https://doi.org/10.1080/01690965.2013.779385

Marelli, M. and Baroni, M. (2015). Affixation in semantic space: Modeling morpheme meanings with compositional distributional semantics. *Psychological Review*, *122*(3):485. https://doi.org/10.1037/a0039267

Matthews, P. H. (1974). *Morphology. An Introduction to the Theory of Word Structure*. Cambridge University Press, Cambridge.

Matthews, P. H. (1991). *Morphology. An Introduction to the Theory of Word Structure*. Cambridge University Press, Cambridge.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Open Science Collaboration (2015). Estimating the reproducibility of psychological science. *Science*, *349*(6251):aac4716. https://doi.org/10.1126/science.aac4716

Pham, H. and Baayen, R. H. (2015). Vietnamese compounds show an anti-frequency effect in visual lexical decision. *Language, Cognition, and Neuroscience*, *30*(9):1077–1095. https://doi.org/10.1080/23273798.2015.1054844

Pirrelli, V., Ferro, M., and Marzi, C. (2015). Computational complexity of abstractive morphology. In Bearman, M., Brown, D., and Corbett, G. G., editors, *Understanding and measuring morphological complexity*, pages 141–166. Oxford University Press Oxford. https://doi.org/10.1093/acprof:oso/9780198723769.003.0008

Plag, I. (2003). *Word-formation in English*. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511841323

Plag, I., Homann, J., and Kunter, G. (2017). Homophony and morphology: The acoustics of word-final S in English. *Journal of Linguistics*, *53*(1):181–216. https://doi.org/10.1017/S0022226715000183

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Seidenberg, M. (1987). Sublexical structures in visual word recognition: Access units or orthographic redundancy. In Coltheart, M., editor, *Attention and Performance XII*, pages 245–264. Lawrence Erlbaum Associates, Hove.

Shafaei Bajestan, E. and Baayen, R. H. (2018). Wide learning for auditory comprehension. In *Interspeech*. https://doi.org/10.21437/Interspeech.2018-2420

Shaoul, C. and Westbury, C. (2010). Exploring lexical co-occurrence space using hidex. *Behavior Research Methods*, *42*(2):393–413. https://doi.org/10.3758/BRM.42.2.393

Spencer, A. (1991). *Morphological Theory: An Introduction to Word Structure in Generative Grammar*. Cambridge University Press, Cambridge.

Stump, G. (2001). *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge University Press. https://doi.org/10.1017/CBO9780511486333

Taft, M. (1994). Interactive-activation as a framework for understanding morphological processing. *LCP*, *9*(3):271–294.

Weaver, W. (1955). Translation. In Locke, W. N. and Booth, A. D., editors, *Machine Translation of Languages: Fourteen Essays*, pages 15–23. MIT Press, Cambridge.

Weingarten, R., Nottbusch, G., and Will, U. (2004). Morphemes, syllables and graphemes in written word production. In Pechmann, T. and Habel, C., editors, *Multidisciplinary approaches to speech production*, pages 529–572. Mouton de Gruyter, Berlin. https://doi.org/10.1515/9783110894028.529

Zwitserlood, P. (2018). Processing and representation of morphological complexity in native language comprehension and production. In Booij, G. E., editor, *The construction of words. Advances in construction morphology*, pages 583–602. Springer.

Address for correspondence

R. Harald Baayen
Department of Quantitative Linguistics
University of Tübingen
Wilhelmstraße 19-23
72074 Tübingen
Germany
harald.baayen@uni-tuebingen.de

Co-author information

Yu-Ying Chuang
Eberhard Karls University
Seminar für Sprachwissenschaft/Quantitative Linguistics
yu-ying.chuang@uni-tuebingen.de

James P. Blevins
Homerton College
University of Cambridge
jpb39@cam.ac.uk