1 Introduction

Large language models (LLMs) such as GPT (Radford et al., 2018) and, more recently, GPT-4 (https://openai.com/gpt-4), Open AI Claude (www.anthropic.com/claude) or LLAMA-2 (https://ai.meta.com/llama/), demonstrate unparalleled performance in text generation and text understanding. They merge lexicon, syntax, discourse, and world knowledge using deep learning networks with billions of parameters that are trained on huge volumes of text that can comprise trillions of words (e.g., Touvron et al., 2023).

Large language models differ from standard conceptualizations of how language works in linguistics, many of which have been deeply influenced by formal grammars and finite state technology. Linguistic theories typically come with a set of assumptions, such as the existence of phonemes, morphemes, words, or the classification of words into parts of speech. They work with a set of elementary representations and rules (or constraints) operating on these representations. Such theories often capture high-level generalizations in a way that analysts experience as providing insight and explanations. This can come at the cost of missing many low-level generalizations. As a consequence, rules typically come with exceptions. These exceptions are then listed in the 'lexicon', which according to some theories is the repository for all information that cannot be accounted for by rules.

Since LLMs lack many of these linguistic assumptions and architectural constraints, all they can do is to exploit as much of the statistical structure – be it high- or low-level – in the training data as is helpful for reducing the loss function they are trained on. Typically, LLMs are trained on predicting the next word given a chunk of previous text as accurately as possible. Additional training details aside, the results of this procedure are models that are surprisingly capable of performing a number of linguistic tasks, not the least of which is to produce text which is syntactically, semantically, and grammatically sound. However, understanding what exactly these models are learning is difficult – partly because this depends on the training regime, model architecture, and chosen hyperparameters, but also because our understanding of LLMs and artificial neural networks is as of yet limited.

There are also other more traditional models which focus exclusively on the statistical structure in the data, such as nearest neighbour methods (Skousen, 1989; Daelemans and Van den Bosch, 2005) or random forests (Strobl et al., 2009; Gries, 2020). These likewise often come with the disadvantage that understanding what a particular model is doing and why can be challenging or even impossible. For instance, large recursive partitioning trees may require many low-level splits that are difficult to make sense of from the perspective of linguistic hypotheses.

It is to be expected that our understanding of many of these techniques, including artificial neural networks and more specifically LLMs, will improve in the future (and it may even have done so by the time you are reading this book, given the leaps forward the field of Natural Language Processing is currently making, see for instance Elhage et al., 2021; Linzen and Baroni, 2021). Nevertheless, the question which arises for the field of linguistics is whether it is possible to combine the demonstratively very powerful approach of machine learning exploiting low- and high-level statistical information in language with more traditional insights from linguistics to further our understanding of language.

This book presents an approach to understanding the lexicon and lexical processing using simple or relatively simple but interpretable machine learning. The representations and processes that together form our knowledge of words are often referred to as our 'mental lexicon'. The Discriminative Lexicon Model (DLM; Baayen et al., 2019) is a computational theory of the mental lexicon. The DLM sets up mappings between high-dimensional numerical representations of form and meaning, applying principles of error-driven learning. This model mostly works with simple linear networks to implement these mappings, but it also offers the possibility to implement mappings with deep networks. Because the model makes use of fine-grained numerical representations of form and meaning, it can benefit to a considerable extent from low-level correspondences between details of form and details of meaning. At the same time, because mappings between form and meaning are kept as simple as possible, the analyst is provided with tools for understanding how form and meaning shape each other.

This book introduces **JudiLing**, a computational toolkit that facilitates building computational models for aspects of lexical processing, using the Julia language. Computational modelling requires many decisions about representations and processes. The book provides detailed step-by-step instructions for how to implement such decisions in Julia. Along the way we introduce important theoretical concept underlying the DLM model and its implementation. As a working example we utilize a dataset of Dutch nouns and verbs (Ernestus and Baayen, 2003), ending by addressing the question of what the computational modelling reveals about final devoicing in Dutch. This worked example is

complemented with case studies from a wide range of other, typologically different languages.

In addition to illustrating how computational models can be set up for lexical comprehension and production in different languages, we clarify how quantitative measures can be extracted from these models that in turn can be used as covariates in regression analyses addressing human lexical processing measures such as reaction times and acoustic durations.

In other words, this book introduces the reader to a linguistic and cognitive theory and its computational toolkit, the **JudiLing** package, designed to facilitate the study of the relation between form and meaning and the consequences of these relations for lexical processing in the mental lexicon.

1.1 The Discriminative Lexicon Model

This book introduces the Discriminative Lexicon Model (DLM). The original formulation of the model (Baayen et al., 2019) made use of very simple mappings known as linear mappings. Its central learning algorithm was therefore referred to as Linear Discriminative Learning (LDL). Since then, the model has been enriched with several additional algorithms for setting up mappings between form and meaning, including mappings that make use of deep learning. As a consequence, we now refer to the model as the DLM. In Chapters 1 through 11, for ease of exposition, we introduce basic concepts and ideas using linear mappings. In Chapter 12, we show how deep neural networks can be used to implement mappings between form and meaning.

The DLM builds on the insight that when words' forms and their meanings are both represented as vectors (see Box 1.1) in high-dimensional spaces, there is considerable isomorphy between these spaces. This isomorphy makes it possible to implement reasonably good mappings between form and meaning by means of simple linear mappings between form vectors and meaning vectors. Practically, this means that one can be transformed into the other by means of very simple computations involving only additions and multiplications. (We shall see that non-linear mappings can offer greater accuracy, but, as mentioned above, in what follows, we first work with linear mappings.)

That is, for now, we model comprehension with a linear mapping from form to meaning and production with a linear mapping from meaning to form. These linear mappings can be described mathematically as implementing multivariate multiple regression and equivalently as using networks with units for form dimensions and units for meaning dimensions, and weighted connections from form units to meaning units (for comprehension) and weighted connections from meaning units to form units (for production). Mathematically, the weights in an LDL network are equivalent to the beta coefficients of a multivariate multiple regression model.

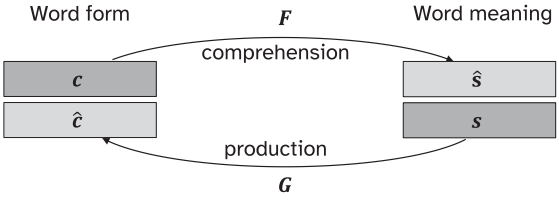

Figure 1.1 The Discriminative Lexicon Model (DLM) models comprehension as a mapping from form **c** to meaning **s** and production as a mapping from meaning **s** to form **c**. Because the mappings are not completely accurate, we denote, borrowing notation from statistics, the semantics predicted by the comprehension mapping by **ŝ** and the form predicted by the production mapping by **ĉ**.

Figure 1.1 illustrates the simple idea of setting up mappings between forms and meanings. In order to obtain the meaning of a word form, a word form's high-dimensional vector representation **c** is mapped to its high-dimensional semantic vector via a comprehension mapping **F**. Because this mapping is generally not completely accurate, we call the predicted meaning **ŝ** (borrowing from notational convention for estimates in statistics). Analogously, in production, a word form's meaning **s** is mapped to the form side via the production mapping **G**. The predicted form vector is called **ĉ**.

As a LINGUISTIC MODEL, the DLM is set up to understand and produce both morphologically simple and morphologically complex words. It is related to realizational morphology (Stump, 2001) in that it is able to model words' forms as a function of their root and morphosyntactic properties. However, the DLM remains agnostic with respect to concepts such as stems and exponents and prefers to work with form units with a much smaller grain size (see Boxes 1.2 and 1.3). As such it is more related to the analogical approach of Word and Paradigm Morphology (Blevins, 2016). As a COGNITIVE MODEL, the DLM generates predictions for lexical processing that can be tested against experimental data. Even though the DLM does not work with discrete units such as stems, exponents, and morphemes, as we shall see, it nevertheless achieves good accuracy both for modelling morphological systems as well as predicting behavioural data.

The following three subsections introduce the DLM from three perspectives: as a linguistic model, as a cognitive model of the mental lexicon implementing error-driven learning, and finally from the perspective of existing models of word recognition and production. The last section of the introduction introduces **JudiLing**, a computational implementation of the DLM facilitating its application as both a linguistic and cognitive analytical tool. This will serve as a starting point for an in-depth presentation and discussion of **JudiLing** and a detailed

guide to how it can be used to address both linguistic and psycholinguistic questions.

Background 1.1 Vectors and vector addition

Informally, for the purposes of this book, we can define a vector as tuple (a 'list' of fixed length) of numbers (for a mathematical introduction to vectors see Deisenroth et al., 2020). For example, column vectors look like this:

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} 0.18 \\ -0.5 \\ 1000.1 \\ 13.4 \end{pmatrix}. \tag{1.1}$$

Similar, row vectors are written in a row:

$$\mathbf{v}_3 = \begin{pmatrix} 0 & 0.5 & 1 \end{pmatrix}. \tag{1.2}$$

A row vector can be transposed to a column vector and vice versa:

$$\mathbf{v}_3^T = \begin{pmatrix} 0 \\ 0.5 \\ 1 \end{pmatrix}, \tag{1.3}$$

$$\mathbf{v}_1^T = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}. \tag{1.4}$$

Conventionally, vector variables are represented with lower case bold letters. Two vectors of the same length can be added by simply adding up elements row-wise:

$$\mathbf{v}_1 + \mathbf{v}_4 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} = \begin{pmatrix} 1+4 \\ 2+5 \\ 3+6 \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \\ 9 \end{pmatrix}. \tag{1.5}$$

1.2 The DLM as a Linguistic Model

As a linguistic model, the DLM provides insight into how both inflectional paradigms and derivational word formation can be modelled computationally. Above, we mentioned that, as a model of word structure, the DLM can be seen as a computational implementation of both realizational (e.g., Stump, 2001) as well as word and paradigm morphology (Blevins, 2016). However, the DLM moves away from representing words' forms as combinations of morphemes, or of stems and exponents (see Boxes 1.2 and 1.3 for more information as to

why). Instead, the DLM views semantics as an inherent and integral part of the morphological system, and builds on the idea that the relations between forms and meanings are at the core of this system. This sets the DLM apart from models predicting inflected or derived forms from other forms, a famous example being the past tense production model of Rumelhart and McClelland (1986a), which derives past-tense forms from present-tense forms (see Section 12.9 and Heitmeier et al., 2021, for further discussion).

Background 1.2 Why no morphemes?

In many standard linguistic theories, phonemes and morphemes are the building blocks of forms. Meanings are often represented by lemmas and functions operating on or features added to these lemmas (such as negation or past-tense inflection). Thus, the word *hands* would be represented at the form level as consisting of two morphemes comprising four and one phonemes respectively

$$[[hand][s]] \tag{1.6}$$

and its semantics would be represented as

$$\begin{matrix} \text{HAND} \\ [+\text{PLURAL}]. \end{matrix} \tag{1.7}$$

Some introductory textbooks to morphology define the morpheme as the smallest meaningful unit of a language (see, e.g., Plag, 2003; Lieber, 2010). Morphemes then constitute the most basic signs, in the sense of De Saussure (1966), linking a minimal unit of form with a minimal unit of meaning. Morphemes are often accepted as a ground truth for theories of language and cognition (see, e.g., Butz and Kutter, 2016; Zwitserlood, 2018). However, within the field of linguistics, most theories of morphology (see, e.g., Stump, 2001; Blevins, 2016) do not make use of the morpheme as theoretical construct because of a wide range of phenomena that run counter to what one would expect if the morpheme, seen as a one-meaning plus one-form unit, would be central to language.

- **Morpheme synonymy**: different form units can realize the same meaning, as for English [s], [z], and [iz] in *cats, hands,* and *houses*.
- **Morpheme homophony**: the same form unit may realize a range of very different meanings, as for the English ending -s: *she walks, the walks, the girl's house, the girls' house, he's here*.
- **Stacking**: one form can realize multiple meanings at the same time. The *o* suffix in Latin *horto* can express both ablative case and singular

number. And to make matters worse, ablative case refers not to one specific meaning, but to a range of meanings that in English would be expressed by prepositions as different as *from, with, by, in*, and *at*.

- **Multiple exponence**: one meaning can be realized in very different parts of a word. For instance, in Classical Hebrew *niktob*, the past tense is indicated by both the form of the pronominal prefix *ni* and the form of the stem *ktob*, the corresponding present tense form is *katab-nu*.
- **Rules of referral**: in inflectional paradigms, sets of forms can build on another form in the paradigm, without inheriting the semantics of that form. For instance, in Estonian, most plural forms are constructed from the partitive singular (*jalgadeks*, translative plural, *jalga*, partitive singular), without the meaning of the partitive singular playing any semantic role.

These phenomena do not fit well with the idea that the basic building block of language is a well-defined form-meaning unit within a calculus in which rules changing forms ('add an *s*') go hand in hand with rules changing meaning ('change number to plural').

Most current morphological theories are therefore 'realizational' theories. They work with units of forms, which can be stems or affixes, or more general, 'exponents'. Importantly, affixes and exponents are conceptualized as expressing or 'realizing' sometimes one but more often several semantic features. The task of the morphologist then is to figure out what the relevant semantic features are, what the relevant stems and exponents are, and to provide rules predicting how the semantic features are realized in form. The resulting morphological grammars (which in some cases are formally equivalent to finite state machines, Beesley and Karttunen, 2003) can provide considerable insight into how a language's morphology works and can also be very useful for second language learners faced with the challenge of mastering complex paradigms such as those of Estonian or Navajo.

Background 1.3 Why no stems and no exponents?

The form vectors that the DLM works with make no attempt at defining stems and exponents. Instead, it replaces these by small sub-lexical units such as diphones or triphones. These sub-lexical units can be stems or exponents, but most such units are not. This design choice offers several advantages.

- It is often far from clear how to segment a word into stems and exponents. By way of example, consider the endings of Dutch verbs in the past tense: *de* for singular subjects and *den* for plural subjects. Is *d* the exponent for the past tense, *e* the exponent for singular number, and *en* the exponent for plural number? Or are the exponents simply *de* and *den*? Splitters and lumpers will come to different conclusions. The DLM does not force such, often arbitrary, choices on the analyst.
- How many words should there be for it to make sense to set up an exponent? Dutch *dievegge*, 'female thief', can be analyzed as *dief* 'thief' and *egge* 'female', but *egge* does not occur in any other Dutch word. Wheeler and Schumsky (1980) provide a detailed discussion of this question for English derivational morphology. Is *el* an exponent in English *runnel, shovel*, and *dotel*? These questions can be avoided simply by working with general sub-lexical units and letting the mappings between form and meaning figure out what the functional load of word-final *el* is.
- One often finds bits of form that are meaningful, without the remainder of the carrying word being a sensible unit by itself. In English, *wh* signals questions in words such as *who, what, where, when,* and *why*, but *o, at, ere, en,* and *y* don't make sense on their own. More in general, phonaesthemes are sounds that occur in often small, but sometimes relatively large, sets of semantically somewhat similar words. English examples are *gl* in *glimmer, glitter* and *glisten*, and *fl* in *flap, fling, flow, flutter, fly, flurry, flail, flinch, flop*, and many others. Experimental research suggests that words with phonaesthemes give rise to very similar processing effects as affixed words (Bergen, 2004; Pastizzo and Feldman, 2009). Mappings operating with fine-grained sub-lexical units can in principle capture these kinds of effects, without requiring decomposition into discrete morphological constituents (see, e.g., Baayen et al., 2011).

By way of example, consider the English verb form *walks*. In terms of realizational morphology (Stump, 2001), the form *walks* realizes, apart from the meaning of the stem, the morphosyntactic properties [SINGULAR], [3RD PERSON], and [PRESENT TENSE]. In the DLM, each of these aspects of the semantics of *walks* can be represented by high-dimensional numeric vectors, as developed in distributional semantics (see, e.g., Landauer and Dumais, 1997; Mikolov et al., 2013). We can add these vectors in order to obtain the meaning vector of the inflected word:

$$\overrightarrow{\text{walks}} = \overrightarrow{\text{walk}} + \overrightarrow{\text{singular}} + \overrightarrow{\text{3rd person}} + \overrightarrow{\text{present}}. \tag{1.8}$$

In what follows, for ease of exposition, we leave out the tense feature.

The DLM is also related to the Word and Paradigm Morphology (Blevins, 2016). Blevins (2016) proposed that morphological systems are entirely word-based, and the only relation existing between different word forms are analogical relations such as

$$\begin{aligned} \text{walk} : \text{walks} &= \text{jump} : \text{jumps} \\ \text{walk} : \text{walked} &= \text{jump} : \text{jumped}. \end{aligned} \tag{1.9}$$

Importantly, such analogical relations are likewise encoded in our vector-addition based semantic representations:

$$\begin{aligned} \overrightarrow{\text{walks}} &= \overrightarrow{\text{walk}} + \overrightarrow{\text{singular}} + \overrightarrow{\text{3rd person}} \\ \overrightarrow{\text{jumps}} &= \overrightarrow{\text{jump}} + \overrightarrow{\text{singular}} + \overrightarrow{\text{3rd person}} \end{aligned} \tag{1.10}$$

Since

$$\begin{aligned} \overrightarrow{\text{walks}} - \overrightarrow{\text{walk}} &= \overrightarrow{\text{singular}} + \overrightarrow{\text{3rd person}} \\ \overrightarrow{\text{jumps}} - \overrightarrow{\text{jump}} &= \overrightarrow{\text{singular}} + \overrightarrow{\text{3rd person}} \end{aligned} \tag{1.11}$$

it holds that

$$\overrightarrow{\text{walks}} - \overrightarrow{\text{walk}} = \overrightarrow{\text{jumps}} - \overrightarrow{\text{jump}}. \tag{1.12}$$

Thus, the difference between $\overrightarrow{\text{walks}}$ and $\overrightarrow{\text{walk}}$ is the same as the difference between $\overrightarrow{\text{jumps}}$ and $\overrightarrow{\text{jump}}$. In other words, when the shift vector $\overrightarrow{\text{walks}} - \overrightarrow{\text{walk}}$ is added to $\overrightarrow{\text{walk}}$ this results in $\overrightarrow{\text{walks}}$, and this shift vector is the same as the shift vector for *jump*.

The idea that the semantics of inflected words are compositional, either in the sense that the meaning of the whole can be derived from the meanings of its semantic features, or alternatively in the sense that the meaning of the whole follows from proportional analogy, is thus straightforward to incorporate in a DLM model. However, a word of caution is in order. There are indications that this way of modelling the meaning of inflected words may lack precision. For example, it has been found that in English, the meaning of a noun plural depends on the semantic class of the noun (Shafaei-Bajestan et al., 2024): shift vectors for plurals are remarkably different for fruits, or animals, or persons. Furthermore, for Russian and Finnish nominal paradigms, the shift vector of the plural has been found to systematically vary with case (Chuang et al., 2022; Nikolaev et al., 2023). In other words, in semantic space, we find interactions between inflectional features and semantic class, or between different inflectional features. The study by Nikolaev et al. (2023) shows that even when many inflectional features are allowed to interact, inflectional forms often resist precise prediction. Interactions between inflectional features are a challenge for any current realizational theory of morphology. In what follows,

we will assume that adding semantic vectors provides a solid first step in the right direction, but this is a simplifying assumption that we will reconsider, see, for instance, Chapter 16.

Whereas the DLM is compositional (or, given the above, fairly compositional) at the level of semantics, it is not a compositional theory at the level of words' forms, in the sense that no attempt is made to build up word forms from stems and exponents, contrary to standard realizational models of morphology. Rather, by default, form is modelled by means of simple overlapping letter/-phone n-grams (see e.g., Baayen et al., 2016b, for theoretical motivation). Furthermore, the relationship between semantics and form is not defined by explicit realizational rules operating on symbolic representations but by means of mappings between form and meaning vectors, using methods from linear algebra.

As an example of what kind of linguistic questions can be answered by means of the DLM, consider the paradigm cell filling problem (Ackerman et al., 2009). To fill a paradigm cell with its form, as a first step, the semantic vector for the pertinent paradigm cell is constructed by adding the semantic vector of the lexeme and the semantic vectors of its semantic features, as illustrated above. As a second step, this vector is mapped to the form side, resulting in a form vector that is used to construct the word form for the paradigm cell (see Figure 1.2, and for further details on the production algorithm, Chapter 8).

Because mappings between the form and meaning side need to be learned (or trained), there can be forms which the model has already seen during the learning phase, and forms that it has not seen before. In more traditional terms, seen forms are paradigm cells which have been encountered before, and unseen forms are 'empty' paradigm cells that speakers have never encountered but that they can nonetheless fill on the fly (unless defective, see Chuang et al., 2022, for Russian nouns). For modelling the filling of empty paradigm cells, the DLM is first trained on all available data. The resulting mapping is then used to predict the form for the empty paradigm cell. So far, the DLM has been used to model both seen and unseen forms in many typologically different languages, such as Latin, English, Estonian, German, Indonesian, Kinyarwanda, Korean, Maltese, Mandarin Chinese, and Russian (see Table 1.1 and for detailed discussion, Chapter 12). Therefore, there is good evidence that, in principle, the DLM can predict previously not encountered forms of a paradigm, and thus provides a solution to the paradigm cell filling problem.

1.3 The DLM as a Cognitive Model

From a cognitive perspective, the DLM is a model of lexical processing. The DLM thus allows to test assumptions about how word forms and meanings might be represented and stored and how their comprehension and production might be learned and unfold during language use. The DLM makes