The Logic of Language: from the Distributional to the Structuralist Hypothesis through Types and Interaction

Juan Luis Gastaldi[a] and Luc Pellissier[b]

[a] GESS, ETH Zürich; [b] LIX, Inria & École Polytechnique

ARTICLE HISTORY
Compiled September 29, 2020

ABSTRACT

The recent success of new AI techniques in natural language processing rely heavily on the so-called distributional hypothesis. We first show that the latter can be understood as a simplified version of the classic *structuralist hypothesis*, at the core of a program aiming at reconstructing grammatical structures from first principles and analysis of corpora. Then, we propose to reinterpret the structuralist program with insights from proof theory, especially associating paradigmatic relations and units with formal types defined through an appropriate notion of interaction. In this way, we intend to build original conceptual bridges between linear logic and classic structuralism, which can contribute to understanding the recent advances in NLP. In particular, our approach provides the means to articulate two aspects that tend to be treated separately in the literature: classification and dependency. More generally, we suggest a way to overcome the alternative between count based or predictive (statistical) methods and logical (symbolic) approaches.

KEYWORDS
Natural Language Processing · Structuralism · Distributional Hypothesis · Structuralist Hypothesis · Type Theory · Realizability · Linear Logic

Introduction

The past two decades have witnessed a remarkable development of the field of Artificial Intelligence (AI). This new AI wave has been brought about by the renewal of the old technique of artificial neural networks, in the form of a family of models taking advantage of the new capacity of increasing the number of hidden layers, making those neural networks *deep*. The expansion of computational power during the last decades and the availability of huge amounts of data resulting from the massive adoption of digital devices by the population, as well as the new role played in research by companies such as Google, Facebook or Tesla, provided the environment and resources necessary to lever the application and use of those formal models to the status of true revolution.

If the undeniable success of the new generation of neural networks is usually acknowledged from a technical and societal perspectives, the scientific and epistemological aspects of such transformation are more difficult to assess. However, the latter are indeed real, both within the fields with which AI is directly concerned (computer

email: juan.luis.gastaldi@gess.ethz.ch luc.pellissier@inria.fr

science, data analysis, mathematics, engineering, etc.) and in those to which it can be directly or indirectly applied. In this sense, the debate around the epistemological import of the surprising results of deep neural networks (DNNs) has been mostly governed by the revived alternative between connectionist and symbolic approaches. Significantly, both perspectives covet the same battlefield of the "human mind" as the object and the source of epistemological enquiry, thus centering the discussion around the validity of DNNs as models of human cognition[1].

Without denying neither the relevance of the event originated by the new AI wave nor its capacity of raising decisive epistemological questions, in this paper we would like to assume a different attitude. Critical of any partisan position and avoiding the false alternative between total acceptance or plain rejection, we propose to take advantage of the impact of DNN techniques in one of the main areas upon which that success is built, namely linguistics and natural language processing (NLP), to assess, not the legitimacy of neural architectures as models of linguistic faculties, but the nature of language itself, as a possible object of formal scientific treatment. By interpreting the significance of the current "deep learning" revolution on the study of natural language as a thesis on the nature of a scientific object rather than of scientific knowledge, we intend to elaborate a critical approach where such a thesis can be pushed further by appealing to theoretical fields whose connection to current state of the art in AI seems remote at best. In this way, we turn what could very well be praised as a revolution by some and dismissed as a hype by others into the occasion for a critical reflection on a traditional object of scientific enquiry and for the establishment of new connections between hitherto unrelated fields. Rather than an attempt to present or evaluate results, the following pages should be read as an exercise of conceptual-bridging, in the hope that they can inspire further work and unexpected outcomes, as well as an alternative perspective on computer science where computational methods are not trapped in the design of imitating and replacing humans.

More concretely, by associating the theoretical basis of DNN linguistic models to classic linguistic structuralism, we intend to propose the first elements of a theoretical framework renewing the structuralist program in the wake of recent NLP advances by reinterpreting one of its key components in the terms of a recent proof-theoretical approach to logical types, based on a notion of interaction between computational processes. Not only can this framework establish original connections between the fields of NLP, classical structuralism and type-theoretical logic, but it can also suggest novel ways in which statistical approaches can be articulated with logical systematizations, thus circumventing the traditional alternative between connectionist and symbolic methods.

In the first section of this paper, we consider the success of DNN models of NLP and propose to associate it to a theoretical stance rather than a technical feat, namely the "distributional hypothesis". We briefly review the history of distributionalism and of its formal implementations, against the backdrop of which DNN models of language should be understood. In Section 2, we suggest that the distributional hypothesis constitutes a corollary and simplified version of a stronger claim, which we call the "structuralist hypothesis", owing to its origin within the framework of classic structuralist linguistics. We then present its focus on paradigmatic derivation as its main conceptual asset with respect to contemporary versions of distributionalism, and elaborate on the advantages that this perspective could represent as a possible strengthened

[1] For a representative instance of this widespread positioning of the AI problem, one can refer to the recent "AI Debate" between Yoshua Bengio and Gary Markus (Montreal, 2019). Cf. https://montrealartificialintelligence.com/aidebate/.

perspective over distributional phenomena. In Section 3 we address some of the main difficulties faced by a possible formal implementation of the derivation of paradigmatic structures and advance that a type theoretical approach, at the crossroads of contemporary logic and computer science, can provide a powerful interpretation of paradigmatic units which could overcome the obstacles in question. In Section 4, we introduce such a formal framework, where types are defined through an original notion of computational interaction stemming from the linear logic program, in a way which is compatible with the stakes previously elaborated. Finally, Section 5 presents some hints on how that framework could be applied for the study of natural language corpora in the direction defined by the structuralist hypothesis. We conclude with a short summary and suggestions for future work.

1. DNNs in NLP, or the Triumph of Distributionalism

When looked at closely, it quickly appears that the recent success of artificial intelligence in the treatment of natural language is not the consequence of computers being more "intelligent" in any sense other than metaphorical. More precisely, the remarkable results exhibited by recent developments in the field of natural language processing (NLP) do not find their source in any successful attempt to explicitly model human faculties, competences or behaviors. Instead, those results are to be attributed to the capacity of a family of algorithms implementing different versions of DNN models to solve a series of tasks associated with the properties of natural language—such as machine translation, question answering, sentiment analysis or summarization—by processing ever-increasing amounts of linguistic data.

Significantly, various network architectures (MLPs, CNNs, RNNs, LSTMs, Transformers, etc.) have been used to treat different tasks, and the increase of performance for a given task has been commonly brought about by the substitution of one architecture by another. Yet, those models differ by significant features—customarily named, still owing to a metaphorical perspective, after cognitive faculties such as "perception" (eg. MLP), "memory" (eg. LSTM) or "attention" (eg. Transformer). This simple fact prevents from attributing to any one of them a decisive epistemic capacity with respect to general linguistic phenomena. However, devoid of the specific mechanisms by which each algorithm organizes the internal representation of the input data, DNN models can only be characterized through a high level strategy consisting in approximating a function through successive layers of distributed representations of a given input, which can compute the expected output for a given task. Unsurprisingly, another cognitive metaphor accompanies this characteristic mechanism of DNNs, namely that of "learning", which remains as insufficient as the others to explain the efficacy of such models in the processing of natural language.

Now, if we take our eyes off their strictly technical aspects and the metaphors that usually surround their epistemic claims, it is possible to see that all those models, insofar as they take natural language as their object, share a unique *theoretical* perspective, known as the *distributional hypothesis*. Simply put, this principle maintains that the meaning of a word is determined by, or at least strongly correlated with, the multiple (linguistic) contexts in which that word occurs (called its "distribution")[2]. As such, a distributional approach is at odds with the generative perspective that dominated linguistic research during the second half of the 20th century. Indeed, the

[2] Cf. Sahlgren (2008); Lenci (2008, 2018); Gastaldi (2020) for an in-depth presentation and discussion of the distributional hypothesis.

latter intends to account for linguistic phenomena by modeling linguistic competence of cognitive agents, the source of which is thought to reside in an innate grammatical structure. In such a framework, the analysis of distributional properties in linguistic corpora can only play a marginal role, if any, for the study of language[3]. By referring the properties of linguistic units to intralinguistic relations, as manifested by the record of collective linguistic performance in a corpus, the distributional hypothesis imparts a radically different direction to linguistic research, where the knowledge produced is not so much about cognitive agents than about the organization of language. Hence, understood as a hypothesis, *distributionalism constitutes above all a statement about the nature of language itself*, rather than about the capacities of linguistic agents.

If the success of DNN models is to be endowed with epistemological significance, it is then as the triumph of this conception of language that it should be primarily understood. Yet, linguistic distributionalism is far from being new. As often recalled in the recent NLP literature, the distributional hypothesis finds its roots in the decades preceding the emergence of generative grammar, in the works of authors such as J. R. Firth (1957) or, more significantly before him, Z. Harris (1960; 1970a). However, it can be argued that this classical work in linguistics was chiefly theoretical. For, although classical distributional methods provided some formal models (by the standards of that time) and even some late computational implementation tests on specific aspects of linguistic structure (cf. Harris (1970b,c)), they were not generally applied on real-life corpora at a significant scale.

And yet, DNN models are not the first to have achieved such formal implementation either: their use of the distributional hypothesis was long preceded by a family of linguistic models whose origins go back to the early 1970s[4]. The main idea of such models—often collectively referred to as *vector space models* (VSMs)—consists in representing linguistic units as *vectors*, whose dimensions are the possible linguistic contexts in which those units occur in a given corpus. The components of those vector representations of linguistic units such as words collect statistical information about the latter's distribution with respect to those contexts. For instance, the j^{th} component of the vector representing the unit i is given by the number of occurrences (or some information-theoretical transformation thereof) of the unit i within that context j. In that way, row or column vectors in a term-context matrix provide a convenient representation for deriving significant properties of linguistic units from their distributional characteristics. In particular, computing the distance between any pair of such vectors amounts to computing their distributional similarity (the more similar the distribution of two units, the smaller the distance of their vector representations) which was thus shown to be directly correlated with different forms of linguistic relatedness.

Within the scope of these matrix models, different configurations of linguistic contexts have been studied[5]. Particularly significant in this sense was the work of Schütze (1992, 1993), introducing the idea of contexts as windows of a given size around terms. Thus, if our corpus contains the first line of Borges's *Two English Poems*[6]:

`The useless dawn finds me in a deserted streetcorner; I have outlived the night.`

[3] Chomsky's rejection of probabilistic methods is well-known, as is his frequently quoted statement that "the notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term" (Chomsky 1969). For an early exposition of this viewpoint, see (Chomsky 1957, § 2.4).

[4] See Turney and Pantel (2010) for an overview.

[5] See Sahlgren (2006) for a historical overview.

[6] All our subsequent examples will be taken from the English language. We introduce the convention of writing linguistic expressions under analysis in a `monospace` font.

the context of the *focus word* `finds` for a window of size ±2 is:

`useless dawn ( ) me in`

while that of `me` is:

`dawn finds ( ) in a`

In this way, linguistic contexts are implemented by means of a sliding window of a fixed size throughout the corpus. As such, it does not seem reasonable to represent those contexts as linguistic units in their own right as, for instance, documents, paragraphs or sentences can be. The proposed solution was then to consider individual words as contexts of focus words, in a word-word matrix M where the cell $M_{i,j}$ collects the frequency in which the j^{th} word in the vocabulary occurs within the context window of the the i^{th} word taken as focus word, throughout the corpus. Notice that, in this way, contexts cease to be treated as explicit linguist units and become just formal means to draw distributional information for focus terms[7]. Finally, a key component of these matrix models is given by the application of dimensionality reduction procedures upon the resulting high-dimensional matrix. Such a reduction relies on classic methods of matrix factorization, and on SVD[8] in particular, which results in words being represented as low-dimensional dense vectors over what can be understood as latent features of the linguistic space represented by the original matrix. By means of this reduced space, the models also increase their capacity of generalization[9].

In more than one sense, DNN models for NLP can be seen as a way of producing and manipulating low-dimensional dense vector representations by other means than those of matrix models. Indeed, in the wake of the first DNN architectures introduced for specific linguistic tasks[10], researchers progressively realized that the initial (or "projection") layer could be considered as producing generic vector representations for the corresponding input words, and could be independently trained accordingly, to be used as the standard input form for DNNs oriented towards different tasks.

In their most elementary form, such neural models for computing word vector representations, or *word embeddings*, associate a random vector of arbitrary but fixed length to each word in a vocabulary, and train those vectors as a hidden layer in a dedicated neural network whose task is to predict words out of the words surrounding them in a given corpus. For instance, in the case of one of the pioneering models, the Skip-gram model of Mikolov et al. (2013), a low-dimensional hidden layer is trained to predict the words contained within the context window of the input words throughout a corpus. The resulting low-dimensional vectors corresponding to those input words encode, in this way, their distributional information, and can then be used as distributed representations for those input words in other DNN architectures dealing with different downstream tasks.

Although produced in a very different way than dense vector representations of previous matrix models, neural word embeddings rely on the same distributional phe-

[7] See Sahlgren (2008) for an assessment of the models based on those two kinds of context forms. As it will become clear in the following sections, our approach differs from Sahlgren's understanding of this distinction, in particular concerning its structuralist interpretation. We have exposed the reasons for this disagreement in Gastaldi (2020).

[8] *Singular-value decomposition* factorizes a matrix M into $M = U\Sigma V$, where V and U represent changes of bases in the origin and arrival spaces in such a way that Σ is diagonal. In other words, it produces a representation of the spaces of words and contexts along so-called *features* such that the action of M is simple to describe.

[9] See, for instance, Landauer et al. (2007) for a comprehensive presentation of *Latent Semantic Analysis* (LSA), one of the most popular models among DSMs.

[10] See Bengio (2008) for an overview of early DNN NLP models.

nomenon. Indeed, it has been shown that such word embeddings encode a great amount of information about word co-occurrence (Schnabel et al. 2015). More significantly, in a series of papers following the introduction of the first neural word embedding models, Levy and Goldberg (2014b) showed that the Skip-gram model was indeed performing an implicit factorization of a (shifted) pointwise information word-context matrix. What is more, the authors were capable of exhibiting performances comparable to that of neural models by transferring some of the latter's design choices and hyperparameter optimization to traditional matrix distributional models (Levy and Goldberg 2014a; Levy, Goldberg, and Dagan 2015).

Since the emergence of those pioneering neural embedding models establishing distributed vector representations as the fundamental basis for the vast majority of DNN NLP models, increasingly sophisticated embedding models have been proposed, which take into account, among others, sub-lexical units (Bojanowski et al. 2016; Sennrich, Haddow, and Birch 2016) or contextualized supra-lexical ones (Peters et al. 2018; Devlin et al. 2018; Radford 2018; Brown et al. 2020). Their architecture and computational strategies differ in multiple ways[11]. However, at their most elementary level, they all share the same simple yet not trivial theoretical grounding, that of the distributional hypothesis, and even akin basic means of setting it up to determine the properties of linguistic units out of the statistics of their contexts in a given corpus.

2. Under Distributions, the Structure!

If we look back to its origins, it is possible to see that the distributional hypothesis constitutes, in fact, a corollary, or rather a simplified and usually semantically oriented version of a classic and more comprehensive and elaborated approach to linguistic phenomena, known as *structuralism*. Structuralist linguistics precedes, and at least in part includes Harris's work, finding its most prominent American exponent in Harris's mentor, L. Bloomfield (cf. 1935), while its European roots go back to the seminal work of F. de Saussure, at the beginning of the 20th century, further developed by authors such as R. Jakobson and L. Hjelmslev (cf. de Saussure (1959); Jakobson (2001); Hjelmslev (1953, 1975)).

As distributionalism, structuralism is above all a theory about the nature of language rather than linguistic agents, based on a series of interconnected conceptual and methodological principles aiming at (and to a great extent required by) the complete description of linguistic phenomena of any sort. All those principles are organized around the central idea that linguistic units are not immediately given in experience, but are, instead, the formal result of a system of oppositional relations that can be established, through linguistic analysis, at the level of the multiple supports in which language is manifested. A thorough assessment of the whole set of structuralist principles falls out of the scope of the present paper[12]. However, it is worth focusing on one of those principles which represents a key component of what structuralism takes to be the basic mechanism of language, namely the idea that those relations constituting linguistic units are of two irreducible yet interrelated kinds: *syntagmatic* and *paradigmatic*.

[11] For a good presentation of the variety of word embedding models, one may refer to Pilehvar and Camacho-Collados (2020).

[12] One may consult Ducrot (1973) for synthetic yet precise and faithful presentation of linguistic structuralism, as well as Maniglier (2006) for an in-depth analysis of its conceptual and philosophical stakes. We have addressed the connection between the structuralist approach and current trends in NLP in Gastaldi (2020).

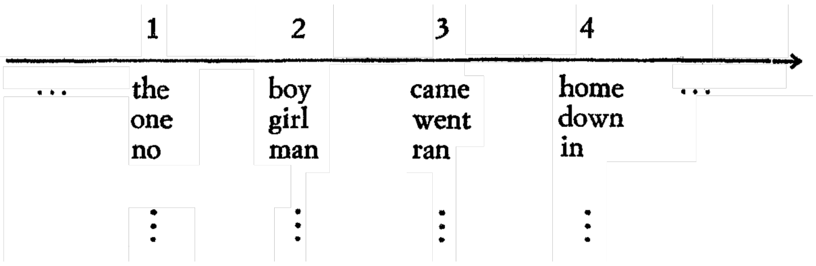


Figure 1. Hjelmslev's illustration of syntagmatic and paradigmatic relations respectively represented by the horizontal and the vertical axes (Hjelmslev 1971, p. 210).

2.1. *Syntagmas and Paradigms*

In their most elementary form, *syntagmatic* relations are those constituting linguistic units (eg. words) as part of an observable sequence of terms (eg. phrases or sentences). For instance, the units `find`, `me` and `in` in our previous example (p. 4) are syntagmatically related. Such units are thus recognized as coexisting in the same linguistic context, bearing different degrees of solidarity. It is this syntagmatic solidarity that contains the essence of the distributional hypothesis, as evidenced by Saussure's words:

> What is most striking in the organization of language are *syntagmatic solidarities*; almost all units of language depend on what surrounds them in the spoken chain or on their successive parts. (de Saussure 1959, p. 127)

Yet, structuralism considers another kind of dependencies that a linguistic unit can contract, namely associative or *paradigmatic* relations with all the other units which could be substituted to it at that particular position. Such units are not—and could not be as such—present in the explicit linguistic contexts of the term being considered. In our example, `me` bears a paradigmatic relation to units such as `you`, `her` or `someone` in the context of `dawn finds ( ) in a`. While syntagmatic relations establish coexisting linguistic units, paradigmatic relations hold between alternative ones, thus implying an exclusive disjunction[13]. Sets of units related syntagmatically are said to form *syntagmas* (or *chains*), while sets of paradigmatically related units constitute *paradigms*.

Figure 1 shows an illustration by Hjelmslev of syntagmatic and paradigmatic relations between lexical units (i.e. words) in the context of a phrase or sentence, with the horizontal axis representing possible syntagmatic relations and the vertical paradigmatic ones. But such relations are not restricted to lexical units, and can be shown to hold between linguistic units at different levels, both supra and sub-lexical. For instance, following another example from Hjelmslev (1953, p. 36), from the combinations allowed by the successive paradigms $\{\mathtt{p},\mathtt{m}\}$, $\{\mathtt{e},\mathtt{a}\}$ and $\{\mathtt{t},\mathtt{n}\}$ we can obtain the words `pet`, `pen`, `pat`, `pan`, `met`, `men`, `mat` and `man`, as syntagmas or chains of a higher level than that of the initial units (characters in this case).

It follows that, from a structuralist point of view, the properties of linguistic units are determined at the crossroads of those two kinds of relations. Now, even without entering into all the subtleties associated with this dual determination of units[14],

[13] Or, anticipating on Section 4, an additive disjunction.

[14] Actually, for structuralism, linguistic or semiological units are determined at the intersection of not one but two sets of such series of syntagmatic and paradigmatic relations: the signifier and the signified, or the expression and the content planes. The need of a second set of syntagmatic-paradigmatic relations can, in principle, be explained by the insufficiency of just one series to determine all the relevant properties of units (the two sets borrowing determinations from one another). For the sake of simplicity, in this paper we restrict

considering paradigmatic relations in addition to syntagmatic ones enlarges the perspectives and goals of the linguistic analysis stemming from a popularized version of distributional semantics. For if, following the distributional hypothesis, the properties of words can be established through the analysis of their linguistic contexts, the paradigmatic structure revealed by the old structuralist lens can provide a much more precise account of the mechanisms involved in the relation between terms and contexts.

2.2. *Syntactic and semantic structural content*

If at any point of a linguistic sequence we can establish the multiple paradigmatic relations at play by providing the specific list of possible units from which the corresponding unit is chosen, a manifold of both syntactic and semantic structural features can be represented. Syntactic, in the first place, as evidenced by the example in Figure 1, where a word like `boy` can be substituted by `girl` or `man`, or even `sky` or `sadness`. But it could not be substituted by `the`, `have` or `from` without making the sentence ungrammatical, i.e. making the sequence of words not be a sentence at all, and hence not belong to a language, in this case English. Such limitation of the domain of possibilities operated by paradigmatic relations at each position of a syntagmatic chain ensures the successful interaction between the terms in that chain, i.e. their capacity to combine in such a way that they constitute a unit of a higher level. The corresponding restrictions respond to multiple dependencies within the syntagmatic chain, recalling that language cannot be reduced to a mere "bag of words". In that sense, they concern above all syntactic phenomena, like the one evidenced by Chomsky's famous example (1957) opposing `Colorless green ideas sleep furiously` to `Furiously sleep ideas green colorless`, where the interaction of the terms in the first case succeeds in establishing a sentence as a linguistic unit of higher level than those terms, while the one in the second case fails. Yet, such dependencies or restrictions do not hold directly between terms but between classes of terms, and are thus difficult to capture explicitly for an analytic procedure focused exclusively in syntagmatic relations. In contrast, the classes established by a paradigmatic viewpoint can contribute to restore those structural syntactic properties explicitly.

As for semantics, there are at least two ways in which paradigms help to specify the way in which distributions contribute to determining the meaning of terms. On the one hand, by being included in a class of substitutable terms, a term receives from the latter a positive characterization, given by the common properties shared by all the terms in the class. In our example, all the words susceptible of occupying the place of `boy` will most certainly not only be nouns (syntactically) but also agents who can come, go and run. Even in the case of unusual substitutions, such as `sky` or `sadness`, the common characteristics of regular terms constituting the paradigm induced at that position by the syntagmatic chain will invest those unusual terms with specific content attributes. If instead of `boy` we had found the term `gavagai` (Quine 2013) at that place, then the corresponding paradigm $\{\texttt{boy}, \texttt{girl}, \texttt{man}, \ldots\}$ would contribute to reducing the complete semantic indeterminacy by projecting upon it the semantic attributes shared by the terms in the paradigm.

However, if this was the only understanding of the meaning of linguistic units provided by paradigmatic relations, then the content of one term would be indistinguishable from that of any of the members of its paradigm. At best, its meaning could only

ourselves to syntagmatic and paradigmatic relations of expressions only, which is also closer to the way in which this problem is treated in NLP.

be singularized at the intersection of all the paradigms to which it belongs. Yet, the mere existence of more than one member in a paradigm is an indication of the fact that the content of those members is not identical, as subtle as that difference might be. From this perspective, the choice of a particular term within the syntagmatic chain is done at the expense of all the others in the corresponding paradigm. Not only is such a choice related to the content of the term, but it can also be understood as constitutive of it. Indeed, following the classic views of Shannon (1948), in line with those of structuralism on this point[15], the content conveyed by a term is completely determined by its choice among a class of other possible terms.

If we agree to call *characteristic content* that component of the meaning of terms that is shared by all the members of a paradigm, we can call *informational content* this other component which singularizes each term by contrast with all the others belonging to the same paradigm. In both cases, the meaning is determined as a result of oppositional or differential relations, but in one case what is differentiated is a group of terms, which thus receive a common content, while in the other singular terms differentiate themselves from that common content to convey a singular meaning. In both cases, we can see how the explicit derivation of paradigms can contribute to specify the way in which linguistic contexts condition or determine the meaning of linguistic units. In the case of characteristic content, no common properties could be positively determined for a term without an explicit class of substitutable terms from which to derive them. As for the informational content, instead of uniformly computing the information conveyed by one term with respect to the entire vocabulary, paradigms restrict at each point of the syntagmatic chain the domain of terms whose distribution is relevant for that computation, thus dynamically taking into account the multiple dependencies between the positions in that chain.

It appears that, through the derivation of paradigmatic relations, the structuralist approach can capture both syntactic and semantic properties of language as the result of one and the same procedure. In this way, it recovers one of the most remarkable aspects of current distributional models, and of word embeddings in particular, which also exhibit this joint treatment of syntax and semantics (Mikolov et al. 2013; Avraham and Goldberg 2017; Gastaldi 2020). But unlike the latter, the structuralist representation of those properties is not limited to elementary probability distributions, similarity and relatedness measures or even clustering methods in the global embedding space. Relying on the derivation of paradigms, the structuralist approach promises to provide a representation of language as a complex system of classes and dependencies at different levels.

2.3. *The Structuralist Hypothesis*

We will focus on the technical aspects of such a system in the following sections. However, it is already possible to identify some important conceptual consequences of the strengthening of the distributional hypothesis through structuralist methods. Starting with the fact that, owing to the specification of the mechanisms by which linguistic context conditions the content of terms, a structuralist approach can dispense with the rather elusive notion of *use* supposed to be somehow reflected in the organization of language. Significantly, while resorting to such a notion of use would imply opening the linguistic model to the study of extralinguistic pragmatic or psychological aspects, the

[15] For a historical connection between the structuralist and the information-theoretical approaches to language, see Apostel, Mandelbrot, and Morf (1957); Jakobson (1967).

remarkable results of current distributional models do not benefit from any substantial contribution from them, other than those recorded in the corpus under analysis. This is not to say that psychological or pragmatical studies are not interesting *per se*, or that the results of current models should not be complemented with such studies, but only that, as a matter of fact, those results do not depend on such investigations. The resort to a notion of use in most of the literature around current (DNNs) distributional models thus remains mostly speculative and ineffective. In line with this situation, a structuralist viewpoint suggests that the source of linguistic content (both syntactic and semantic) is to be sought, not in pragmatic or psychological dimensions beyond language, but primarily in the fairly strict system of interdependent paradigms derivable, in principle, from the explicit utterances that system is implicitly governing. As Harris puts it:

> The perennial man in the street believes that when he speaks he freely puts together whatever elements have the meanings he intends; but he does so only by choosing members of those classes that regularly occur together, and in the order in which these classes occur. [...] the restricted distribution of classes persists for all their occurrences; the restrictions are not disregarded arbitrarily, e.g. for semantic needs. (Harris 1970a, pp. 775-776)

It follows that the analysis of a linguistic corpus, inasmuch as it succeeds in deriving the system of classes and dependencies that can formally account for the regularities in that corpus, is a sufficient explanation of everything that is there to be *linguistically* explained. This idea constitutes a key component of what can henceforth be called the *structuralist hypothesis*, namely that linguistic content (including essential aspects of linguistic meaning) is the effect of a virtual structure of classes and dependencies at multiple levels underlying (and derivable from) the mass of things said or written in a given language. Accordingly, the task of linguistic analysis is not just that of identifying loose similarities between words out of distributional properties of a corpus, but rather this other one—before which the latter appears as a rough approximation—of explicitly drawing from that corpus the system of strict dependencies between implicit linguistic categories. If we agree to adopt Hjelmslev's terminology and call *process* a complex of syntagmatic dependencies and *system* a complex of paradigmatic ones (Hjelmslev 1975, p. 5), then the following passage from Hjelmslev's *Prolegomena* can be reasonably taken to express the essence of the structuralist hypothesis:

> *A priori* it would seem to be a generally valid thesis that for every *process* there is a corresponding *system*, by which the process can be analyzed and described by means of a limited number of premisses. It must be assumed that any process, can be analyzed into a limited number of elements recurring in various combinations. Then, on the basis of this analysis, it should be possible to order these elements into classes according to their possibilities of combination. And it should be further possible to set up a general and exhaustive calculus of the possible combinations. (Hjelmslev 1953, p. 9)

Notice that, in Hjelmslev's view, the ultimate goal of linguistic analysis goes beyond the pure description of the data, and pursues the derivation of an exhaustive calculus. This goal is at least partially fulfilled by current distributional models, which are intended to be applied to data outside the corpus upon which they were trained. But if a calculus is necessarily at work in these models once they are trained, so that they can achieve the generalization required to treat previously unobserved data, the principles of that calculus remain entirely implicit. Here too, we can see how the structuralist derivation of a (paradigmatic) system out of (syntagmatic) processes can contribute to providing an explicit representation of such a calculus, based on the particular way in which generalization is achieved trough paradigms. The example in

Figure 1 can offer a first intuition of this mechanism. If, out of the three sentences corresponding to the three horizontal lines of the table, we are able to derive the four paradigms A, B, C, D, corresponding to the latter's columns, and then establish some of their combinatorial properties, for instance, the capacity of composing them in the the order $A \times B \times C \times D$,[16] then the explicit calculus that starts to be drawn in this way appears as the correlate of the generalization achieved by considering all possible combinations of the members of the paradigms at their corresponding positions (eg. `the girl ran home`), the vast majority of which was not present as such in the original data upon which the system was built.

Incidentally, the program attached to the structuralist hypothesis challenges the classic distinction between connectionist and symbolic methods and its philosophical consequences (cf., for instance, Minsky (1991)). While beginning with combinatorial properties of linguistic units as raw data whose structure is only presupposed, the structuralist hypothesis aims at the reconstruction of an explicit and interpretable representation of the structure underlying such data, taking the form of a symbolic system at different levels (from the phonological all the way up to the grammatical or even stylistic level). From this perspective, symbolic systems implementing different aspects of algebraic structures are the direct result of the interaction of terms (including sub- or pre-symbolic ones) reflected in the statistics of given corpora. Conversely, when those symbolic systems are put into practice—in the performance of linguistic agents, for instance—the corresponding symbolic processes cannot but reproduce the statistical properties of the terms upon which that system was derived. Hence, from a structuralist approach, connectionist and symbolic properties appear as two sides of the same phenomenon.

3. Structures and Types

With the rather frail means of the epoch, the classic structuralist approach was able to prove its fecundity in the description of mainly phonological and morphological structures of multiple languages. However, empirical studies of more complex levels of language, and of grammar in particular, received mostly circumscribed and limited treatment. More generally, despite some valuable early efforts (Hjelmslev 1975; Harris 1960) structuralist linguistics encountered difficulties in providing effective formalized methods to describe syntactic structures in their full generality. The rise of Chomsky's generativist program in the late 1950s pushed the structuralist approach into obsolescence, until some of the latter's intuitions were recovered in the form of distributional methods by the resurgence of empiricist approaches in the wake of the emergence of new computational techniques in the 1980s (cf. McEnery and Wilson (2001); MacWhinney (1999); Chater et al. (2015)).

As it turns out, the resurgence of distributional methods has been mostly driven by semantic concerns. However, the success of distributionalism through DNN models goes far beyond semantic properties, covering a wide spectrum of linguistic phenomena, from syntax to style, in such a way that it exceeds the modest claims of the distributional hypothesis, calling for a stronger conceptual foundation like the one suggested by the structuralist hypothesis. Indeed, current models seem to capture a significant amount of structural features of language out of distributional properties, making them available for their precise application to a vast range of downstream

[16] Of course, this example has only the value of an illustration. The analysis of real corpora renders this task far more difficult.

tasks, which tends to confirm the claims of the structuralist hypothesis that those features can be derived from the analysis of linguistic performance alone, contributing to a better grasp of linguistic meaning.

Certainly, the vast majority of recent DNN models that could motivate the reassessment of the structuralist hypothesis do not provide an explicit account of those implicit structural features. In the past years, however, several studies have focused on finding evidence of the fact that those structures are indeed encoded in the resulting models (Linzen, Dupoux, and Goldberg 2016; Enguehard, Goldberg, and Linzen 2017; Dinu et al. 2018; Blevins, Levy, and Zettlemoyer 2018; Goldberg 2019; Clark et al. 2019; Hewitt and Manning 2019; Manning et al. 2020; Bradley 2020). While those studies allow to confirm the idea that DNN models succeed in capturing implicit linguistic structure from purely distributional properties, the question of a method—be it neural or not—that could provide an explicit representation of such structure remains open.

3.1. *Obstacles to Paradigm Derivation*

We have seen the main strategy the structuralist approach proposed to tackle this problem was the derivation of paradigmatic units. But establishing paradigms is a highly challenging task outside strongly controlled and circumscribed conditions. For if, at first sight, paradigms appear as simple classes of terms, such classes have the particularity of being at the same time of an extreme *precision*—since the inclusion of one incorrect term would be enough to jeopardize the successful interaction of linguistic terms—and perfectly *general*—since paradigms potentially contain an indefinite number of terms, either unseen in the data upon which they were derived or even not yet existent in the language under analysis, thus virtually allowing for an indefinite number of syntagmas or linguistic processes.

Those two conditions are somewhat in tension: generality excludes any purely extensional definition of paradigms, while precision makes intensional or logical definitions particularly complex, especially considering that they are to be drawn exclusively from distributional properties. Indeed, such precision is the result of the simultaneous action of multiple restricting principles, which are realized by terms interacting within a definite context. In the expression `one girl has gone`, for instance, the paradigm of `has`, which contains the terms `has`, `had`, `is`, `was`, is delineated simultaneously by `gone`, selecting possible auxiliary verbs, and by `one` and `girl` selecting verbs or verb phrases that are singular.

The situation exhibited by that simple example already allows us to indicate three major obstacles the derivation of paradigms face. The first and most important of those obstacles concerns the nature of the dependencies upon which a paradigm is supposed to be established. As we have seen, the restricting principles through which a paradigm is to be established correspond to several dependencies within the syntagmatic chain. But we have also seen that such dependencies do not hold directly between terms but between classes of terms. From the point of view of paradigmatic derivation, this means that, while we can only rely on terms within the syntagmatic chain—the only ones accessible to experience[17]—, paradigms do not contract dependencies with terms but with their respective characteristic contents (for instance, with the noun and singular characters of the term `girl`). Significantly, the characteristic content of a term

[17] In the words of Saussure (de Saussure 1959, p. 123), only syntagmatic relations are *in praesentia*, while associative or paradigmatic ones are *in absentia*. However, this does not imply that terms are immediately given to experience in the syntagmatic chain. See Footnote 18 below, as well as Maniglier's (2006) fine analysis of this difficult question.

can only be established through its paradigmatic relations, which rely, in turn, upon the same kind of dependencies with paradigms in the context, including the original paradigm we were intending to establish. Indeed, the singular character of `girl` is nowhere to be found other than in the fact that, within this context, terms like `boy` or `man`, but not `boys`, `men` or `girls`, belong to the paradigm of `girl`, a circumstance that depends, among other things, on the fact that the context of the paradigm including `has`, `had`, `is` and `was`, but not terms like `have` or `were` is present in the context. The circularity of the task is manifest: paradigms are needed to establish paradigms. Yet, this circularity is not to be attributed to the method itself, but to the very nature of its object. Indeed, from a purely internal or empirical viewpoint, what are, for instance, adjectives, other than a particular class of terms that appear at the side of nouns? And what are nouns if not something that can accept adjectives at their side? These mutual dependencies are by no means restricted to syntactic classes, but pervade all levels of language: phonological (eg. vowels and consonants), morphological (eg. verb stems and inflections), semantic (eg. agent and actions), or even stylistic (eg. formal and familiar).

A second difficulty defying paradigmatic derivation concerns the composite organization of the restrictions delineating a paradigm. In the example above, for instance, the context of the term `has` most certainly allows drawing a paradigm including `has`, `had`, `is`, `was` and excluding `have`, `are`, `were`. But the interaction of this paradigm with the context is not homogenous and unidimensional. We mentioned at least two different features of that interaction: one with (certain aspects of) the characteristic content of `gone` and the other with that of `girl`. However, if the context is considered as a unanalyzed whole, as it usually is in current distributional methods, those dimensions remain indistinguishable. Considering the different paradigms composing that context (if the difficulty stated above was somewhat overcome) could certainly help, but it would not be enough. For it would still be necessary to know how those paradigms interact with each other establishing a complex system of dependencies. The singular character determining `has`, `had`, `is`, `was`, for instance, may find its source in the paradigm of `girl`, but also in that of `one`. Nevertheless, this would no longer be the case for a sentence like `one day the girl was gone`, where `one` and `girl` interact in a different way than in the original example. If we recall the idea of successful interaction of the previous section, we can say that in the first case both units successfully interact, constituting a unit of a higher level, which in turn contributes, as a new unit, to the definition of the paradigm of `has`, while in the second they only interact indirectly (after interacting with other units) in such a way that the paradigm of `one` does not affect the definition of that of `has`. Into this kind of difficulties fall also classic examples such as `the boy and the girl have`, where the paradigm of `have` should contain plural terms and exclude singular ones, even if none of the paradigms for the words in the context can be expected to explicitly exhibit that characteristic (Chomsky 1957, § 5.2-3). To deal with this difficulty, some sort of compositionality principle should be found. But the mere composition of paradigms is not enough either. A more subtle mechanism is needed to assess the multiple ways in which different compositional principles are capable of interacting to derive a hierarchical structure.

Finally, once a paradigm is established, what constitutes its unity might not be immediately evident from the list of terms it contains. This can be, of course, the consequence of counting on only partial information for its derivation. But the lack of paradigmatic unity can arise even when this is not the case. Take, for instance, Hjelmslev's example in Figure 1. What could possibly be the unity of the paradigm delineated by the terms in the fourth column $\{$`home`, `down`, `in`$\}$, to which one could perfectly imagine to add others such as back, up or yesterday? It appears that, even if its members are not drawn in a purely random way, the internal coherence of such a paradigm is not completely guaranteed by the context, requiring further specification. And indeed, a quick inspection of those members suggests that the paradigmatic class could be analyzed into different subclasses, such as adverbs and prepositions. While the previous difficulty we pointed out can be understood as concerning syntagmatic relations between paradigmatic units defining the structure of linguistic contexts, in this case we are confronted with the problem of the paradigmatic relations between (sub-)paradigms defining the structure of a paradigm containing them. The difficulty of this task resides in that, in principle, the context upon which the paradigm was derived in the first place has no explicit means to perform further discriminations within that paradigm, and it is not clear what the source of those discriminations could be.

These difficulties have not remained unnoticed, even in the old days of structuralist research (see, for instance, Chomsky (1953)). They are also not the only ones that derivation of paradigms can encounter[18], most of all considering we have only presented them though extremely simple illustrations. Real life analysis can only make this situation worse. In particular, trying to perform paradigmatic derivation exclusively through corpus analysis can raise new difficulties unforeseen to a pre-computational structuralist perspective, for which the automatic processing of corpus of significant size remained after all a promising but peripheral possibility. Indeed, most of the structuralist original theoretical and methodological constructions are consciously or unconsciously conceived on the basis of linguistic data that can be produced by elicitation from an informant, if not through simple introspection. Problems like adequacy of probability measures, scarcity of data or impossibility statements (establishing, for instance, that two terms cannot stand in a given relation) barely appear among its original theoretical concerns.

And yet, given the resurgence of distributional methods and the growing necessity of making explicit the mechanisms directly or indirectly responsible for that success, it seems worth restating those main difficulties concerning paradigmatic inference, in the perspective of the renewal of those methods in the new setting. For the structural features current models have been shown to grasp, if only implicitly, are an indication that such difficulties can be overcome. It would not be too audacious to interpret the recent developments of NLP distributional methods, from LSA all the way up to BERT and GPT, as ways of tackling some aspects of those same problems. For instance, the co-determination between terms and contexts seems to be addressed through factorization techniques such as SVD in matrix models, while NN word em-

[18] In particular, we have disregarded here a fundamental problem which is nevertheless central from a structuralist standpoint, namely that syntagmatic relations between terms upon which our construction of paradigms relies as a given, do in fact require to be established in a way that also depends on the paradigmatic relations they are supposed to help constructing. For if at a specific point in the syntagmatic chain we can establish that a split can be observed so that the two units at the sides of that split can be taken to stand in a syntagmatic relation, it is neither because of some substantial and pre-existing nature of words nor due to the unescapable action of whitespaces or punctuation characters (which constitute a rather recent introduction in writing and have no real correlate in speech). If splits can be established at certain points of the syntagmatic chain it is, rather, because paradigmatic relations at both sides of those points allow to decompose the entire syntagmatic unit into subunits (de Saussure 1959, 2, §VI). In Saussure's words: "...spatial [i.e. syntagmatic] co-ordinations help to create associative [i.e. paradigmatic] co-ordinations, which are in turn necessary for analysis of the parts of the syntagm." (de Saussure 1959, p. 128). Hence, in order to be entirely faithful to the structuralist perspective, a segmentation procedure should make part of the derivation of a linguistic system, not just as a preliminary step (like "tokenization") but on a par with paradigmatic derivation. For the sake of simplicity, in this paper we have decided to focus on the challenges of paradigm derivation alone, relying on the lexical segmentation of texts which is standard in most NLP approaches.

bedding models, like word2vec, seem to deal with it through the simultaneous training of word and context vectors (the latter being either combined with the former into one single representation or simply discarded). Similar interpretations could be made, for instance, of attention mechanisms as a way of capturing the distinctive action of different context units upon focus terms, or of contextual embeddings and their capacity of discriminating different alternative meanings blended under the same apparent unit. It is not the place here to carry out those interpretations[19]. It is important, however, to avoid considering the obstacles we have described as indisputable evidence for the impossibility of explicit paradigm derivation. Instead, the assessment of those obstacles should lead to a positive characterization of that task.

3.2. *Paradigms as Types*

Now, if other than negative conclusions are to be derived from those obstacles, we can see that all of them convey the idea that paradigmatic units, as the relevant classificatory units to be derived behind explicit terms, do not pre-exist the dependency relations they contract with other units of the same kind. Hence, if we want to represent the implicit paradigmatic system, it is important to adopt a framework where the dynamic establishment of dependencies is constitutive of its elementary classificatory objects. For this reason, we propose to represent paradigms as *types*.

The notion of type has a twofold meaning owing to its double genealogy (Kell 2014; Martini 2016). On the one hand, it refers to *logical types*, originating with B. Russell and reworked by A. Church in the field of symbolic or mathematical logic. On the other hand, it conveys the idea of *data types*, belonging to the tradition of engineering and programming languages. In the first sense, a type is associated to a restriction, generally corresponding to a rule, aiming at limiting the expressive capabilities of a formal language (and of quantification in particular) in such a way that only "meaningful" expressions (generally intending: non-paradoxical) can be formed. Typically, the purpose of types in this acceptation is to prevent a (propositional) function from taking itself as its own argument, thus avoiding a certain kind of logical paradoxes. To that end, functions can be declared, for instance, to be of a different type than their arguments, and can themselves only be arguments of functions of a different ("higher") type. This results in a classification principle for expressions, but such classification does not intend to characterize those expressions positively (i.e. as having having this or that characteristic) but only with respect to their capacity to interact with others in such a way that the interaction does not result in undesired behaviors.

Data types, in contrast, concern primarily abstraction mechanisms for data manipulation (eg. storing, reading, operating on, etc.) within programming languages. Following Kell (2014), abstraction in this context is to be understood in two related senses. On one hand, abstraction concerns *generalization*, by selecting specific significant features shared by different expressions while discarding others. The numeric expressions `1`, `2`, `3`, for example, refer to singular objects or individuals which have some common features (eg. addition of `1` can be performed on all of them) while other features are specific to only some of them (eg. being odd or even, or resulting in `4` when `1` is added to them). In this way, we can create a more general kind of entity which characterizes those individuals only through their common features (i.e. abstracting

[19] In Gastaldi (2020), we have interpreted embedding methods as ways of dealing with the problem of the (bi-)dual mechanism constitutive of linguistic units. The bi-orthogonality typing presented in the next sections can be understood as an attempt to provide a formal version of that idea.

away all the other specific traits). "Type" is then a generic name for those general entities. Yet, the latter can be given specific names, in such a way that they can now be referred to within the programming language (for instance **int**, as "integer", for `1, 2, 3` in our example). Hence the second aspect of abstraction, namely that of a referential device which allows to talk about individuals without needing to engage with their specific characteristics or even without requiring any of those individuals to be explicitly realized or computed (as when one says "let x be an integer"). It follows that, as data types, types are essentially "named interpretations to which we can refer" (Kell 2014, p. 231)[20]. In contrast to their logical counterpart, data types positively characterize the elements corresponding to them through a common behavior specified by the features that define the type. For instance, establishing that `1, 2, 3` are all of type **int** implies that they all behave in the same way, namely as integers. One has only to think of other common data types in programming language, such as *booleans*, *strings* or *lists*, to get an idea how data types convey a positive characterization of the content of their respective terms.

While those two dimensions underlying the meaning and use of types as formal objects do not necessarily coïncide, they are far from being incompatible. And significantly, they perfectly encompass the two aspects characterizing how paradigms determine the content of natural language units. Indeed, as we have seen in the preceding section, paradigms, as much as types, fulfill the double function of restricting the interactions of terms in such a way that the correctness (or grammaticality) of linguistic processes can be assured, and of positively characterizing the content of those terms by defining a class from where common characteristics can be abstracted and with respect to which the choice of terms becomes meaningful. Therefore, in first approximation, types appear as an adequate way of formally representing the fundamental units of linguistic systems. But more importantly, their twofold understanding harbors a possible response to the problem of defining elementary units as the simultaneous effect of dependent interaction. For, in its evolved logical interpretation, the doctrine of types is explicitly oriented to turn the multiple limitations on the interaction of terms into a system of logical dependencies, while from the programming perspective, the units supposed to lie at both ends of such dependency relations are given a positive purport by associating them to the characteristic behavior of concrete terms. Hence, if those two aspects underlying the meaning and use of formal types could be thought to correspond to each other as two sides of the same procedure, then the open problem of a structuralist analysis of linguistic corpora, and maybe even that of the logical import of statistically inferred features, could be addressed afresh.

The idea of representing linguistic phenomena through types is not new. The first type-theoretical framework for natural language dates back to as early as 1958, with the seminal work of J. Lambek (1958), at the origin of the long tradition of Categorial Grammars, which constitutes a current active area of research (McGee Wood 1993; Moot and Retoré 2012). However, resulting from the crossbreed of the logical tradition of types and the generativist program in formal linguistic, distributional approaches where type systems could be drawn from the unsupervised analysis of corpora received little attention within this framework. The recent success of DNN models has, nevertheless, motivated several attempts to use types to provide explicit representations for both semantic and syntactic structures of language, based on current techniques, and of embeddings and vector representations in particular. Thus, for instance, type

[20] Interestingly, the author also associates data types to specific restricted alphabets from which symbols are drawn during a communication process as understood by information theory (Kell 2014, p. 228).

derivation and typing techniques have been applied to achieve an explicit representation of semantic types, either improving or introducing tasks like fine-grain and hierarchical classification, feature extraction, entity linking, parsing, co-reference resolution or entailment[21]. Other works, in turn, have proposed to combine categorial approaches to neural embeddings in such a way that the powerful framework of logical types can be applied to word vector representations based on the latter's algebraic properties (see for instance Coecke, Sadrzadeh, and Clark (2010); Clark et al. (2016); Coecke (2019); Wijnholds and Sadrzadeh (2019)). The rapid development of this line of research is a promising sign for a type-driven approach to distributional phenomena. However, in the vast majority of the cases, a separation between both dimensions of types persists. Thus, while in the cases in which atomic (mostly semantic) types are constructed through distributional properties the logical dependencies between them are hardly addressed as such, the models mobilizing the full logical capacities of type systems tend to consider atomic types as given, associating them to distributional units through extrinsic means.

Unrelated to the treatment of natural language, a gradual but consistent confluence of the two traditions behind the notion of type has been underway since the 1970s. Triggered by the publication of the Curry-Howard isomorphism, establishing the direct correspondence between computer programs and logical proofs (cf. Groote (1995)), this confluence gave rise to a series of research programs, such as constructive type theory, linear logic or homotopy type theory, developing an intimate connection between computational processes and the conditions underlying logical inference, including the association between logical and data types. As we will show in the next section, the interpretation of some orientations of this tradition in the framework of the analysis of natural language can offer interesting tools to provide an original type-theoretical setting for the reassessment of the structuralist hypothesis.

4. Types, interaction, and orthogonality

Among the traditions stemming from the Curry-Howard correspondence, a singular research program originating in French proof-theory (Girard 1989, 2001; Krivine 2010; Miquel 2020) brings to the fore a notion of *interaction* upon which the types of a type system can be built from an intricate web of dependencies[22]. This original perspective at the intersection of computational processes and logical structures offers a powerful framework to address the difficulties associated with the derivation of paradigms as something more than simple classes, since not only derived types, but also atomic ones can be conceived as resulting from the same procedure[23]. In this section we will present the essential aspects of the construction of abstract types through interaction by means of examples within a simple formal setting, before suggesting, in the last section, how this framework could be used for the analysis of natural language corpora.

One of the canonical ways of representing computational processes in theoretical computer science is the *simply-typed λ-calculus*, an abstract programming language centered around the notion of function and of their application to arguments. Thus,

[21] For an example of some of these works, see Choi et al. (2018); Chen, Chen, and Van Durme (2020); Lin and Ji (2019); Raiman and Raiman (2018); Krishnamurthy, Dasigi, and Gardner (2017); Abzianidze (2016).

[22] The authors are currently writing an article on these developments and how they affect fundamentally the traditional notions of logic.

[23] Indeed, in Girard's *Ludics* (Girard 2001), a tentative reconstruction of the whole of logic from an interactive point of view, there are no atomic types *per se*: atomic types are just types that are not yet decomposed.

the language only knows of two constructions: one defining a function (called "abstraction") and one evaluating a function at an argument ("application"). Moreover, as simply typed, the language assumes the primitive distinction between *terms*, which are functional syntactic representations of computational processes or programs, and *types*, which are logical characterizations of their systematic behaviors.

The relationship between terms and types is expressed through *typing judgments*, which are expressions of the form:

$$\vdash t : A$$

stating that the term t has type A. The formal meaning of the judgment is associated with typing rules (differing from one system to another), and the intended, informal, meaning of the judgment varies depending on the type at the position of A. For instance, if instead of A we have the composite type $B \to C$ [24], the intended meaning of the judgment $\vdash t : B \to C$, expressing that a program t has the type $B \to C$, is that, given any other program u of type B, the computation tu (interpreted as the evaluation of t, viewed as a function, on u, taken as its argument) can proceed and eventually yield a term of type C.

From this elementary example, we can see that types are here associated to the interaction between programs, as expressed through functions. In particular, the intended meaning of types correspond to desired or expected results of that interaction. This general intuition of successful interaction can be rephrased in a vocabulary coming from linear logic (Girard 1989) that we will adopt here:

- two sets A and B of terms are said *orthogonal*[25] if all the elements of A interact successfully with all the elements of B—which we write $A \perp B$;
- given a set of terms A, its *orthogonal* $A^{\perp}$ is the set of all the terms that interact successfully with all the terms in A—in other words, the biggest set orthogonal to A;
- a set of terms that is the orthogonal of another set is said to be a *type*. Indeed, such a set is defined as the set of terms interacting successfully in the same way with respect to a testing set. Moreover, every set that is a type is also closed by biorthogonal (that is, every type A satisfy $A^{\perp\perp} = A$) and this property actually characterizes types. So, the property of being a type can be seen as a closure property—relative to the interaction.

So, in this view, types are defined out of a notion of orthogonality between sets, itself capturing successful interaction between terms. It follows that not every arbitrary set is a type. As a consequence, even atomic types have a structure, which depends on the notions of interaction and of success chosen.

This idea will become clearer through the following examples, closely related to one another.

Example 4.1. Consider a simple programming language whose expressions are limited to integer numbers (written 1, 2,…) and arithmetic functions that add to their argument a fixed integral offset (written +1, +2,…).Their interaction is defined according to the following cases:

[24] That is: a type built from two arbitrary types B and C, through the connective $\to$.

[25] Originally, this vocabulary comes from linear algebra and the orthogonality of linear forms with vectors. We won't have any need of this interpretation.

- two numbers interact by yielding an error:

$$(1)(3) \rightsquigarrow \square$$

- a number and an offset interact by adding the offset to the integer, yielding an integer:

$$(+3)(2) \rightsquigarrow 5$$
$$(8)(+2) \rightsquigarrow 10$$

- two offsets interact by adding their offsets, yielding an offset:

$$(+6)(+3) \rightsquigarrow +9$$

This interaction is commutative: the order of the terms is of no consequence on the result.

Suppose that we consider offsets as partial computations, and hence only consider numbers as results: it is then natural to say that an interaction is successful if it ends in a result, i.e. a number. So, we define the orthogonality as: for two terms a and b, $a \perp b$ if the result of ab is a number.

We immediately see that for any arbitrary set of numbers, including singletons, the orthogonal is the entire set of offsets: for instance, $\{1\}^{\perp} = \{4, 7\}^{\perp} = \{+0, +1, \dots\}$. Indeed, only offsets interact with numbers by producing a value. In the same way the orthogonal of any set of offsets (say, +3 and +5) is the set of numbers: $\{+3, +5\}^{\perp} = \{0, 1, \dots\}$. We can write this diagrammatically, with a dashed arrow representing the orthogonal:

$$\begin{tikzcd}
\{1\} \arrow[r, dashed, bend left] & \{+0, +1, \dots\}
\end{tikzcd}$$

$$\begin{tikzcd}
\{+3, +5\} \arrow[r, dashed, bend left] & \{0, 1, \dots\}
\end{tikzcd}$$

By combining these two observations, we see that the *biorthogonal* of any set of numbers is the set of *all* numbers: $\{1\}^{\perp^{\perp}} = \{4, 7\}^{\perp^{\perp}} = \{0, 1, \dots\}$ and the biorthogonal of any set of offsets is the set of all offsets: $\{+3, +5\}^{\perp^{\perp}} = \{+0, +1, \dots\}$. So, to complete our diagram:

$$\begin{tikzcd}
\{1\} \arrow[r, dashed, bend left] & \{+0, +1, \dots\} \arrow[d, dashed, bend left] \\
\{+3, +5\} \arrow[r, dashed, bend left] & \{0, 1, \dots\} \arrow[u, dashed, bend left]
\end{tikzcd}$$

Finally, we also remark that no term interacts correctly with both a number and an offset, and that the orthogonal of the empty set is the set of all terms. From these considerations, we can build a lattice containing all the types in this language, represented in Figure 2.

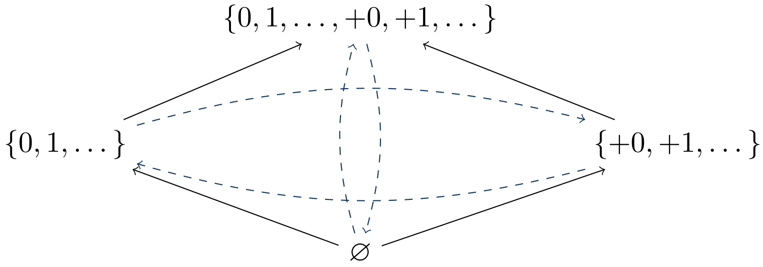


Figure 2. The lattice of types of numbers and offsets.

As simple as this example may be, it already shows how a notion of successful interaction (that is, the specification of the interaction and of what counts as a result), allows to build, at once, different atomic types, characterizing the behavior of terms, and relations between those types: *orthogonality*, representing a form of compatibility in the interaction between two types, but also an incipient form of *subtyping* (for instance, between the type of all terms, and the types of numbers or of offsets), representing different degrees of specificity.

Example 4.2. Example 4.1 can be refined. Suppose now that not all numbers are considered suitable results, but only those that are small enough, say, less than 9. In that case, we define the orthogonality to be: for two terms a and b, $a \perp b$ if the result of ab is a number in $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$.

We immediately see that the structure of types for this newly defined orthogonality relation is much more complex. Indeed, consider the set containing the number 1. Its orthogonal consists of the offsets smaller or equal to +7:

$$\{1\}^{\perp} = \{+0, +1, +2, +3, +4, +5, +6, +7\},$$

because, for greater offsets, the result will not be less than 9 (for instance $(+10)(1) \rightsquigarrow 11$). In contrast, the orthogonal of $\{8\}$ is

$$\{8\}^{\perp} = \{+0\}.$$

Moreover, the orthogonal of a set containing multiple elements is determined by its greatest one: indeed, everything that interacts well with the greatest interacts well with all the others; so, for instance, we have:

$$\{4, 5, 7\}^{\perp} = \{+0, +1\}.$$

Finally, for every set containing an integer greater or equal to 9, the orthogonal of its singleton is empty, as no small integer can be computed from them.

The same phenomenon appears for orthogonals of offsets, so in particular,

$$\{7\}^{\perp\perp} = \{+0, +1\}^{\perp} = \{0, 1, 2, 3, 4, 5, 6, 7\}.$$

From the point of view of the interaction chosen here, all the terms in $\{0, 1, 2, 3, 4, 5, 6, 7\}$ act as 7 would.

The entire lattice corresponding to this example, much finer than the precedent one, is presented in Figure 3.

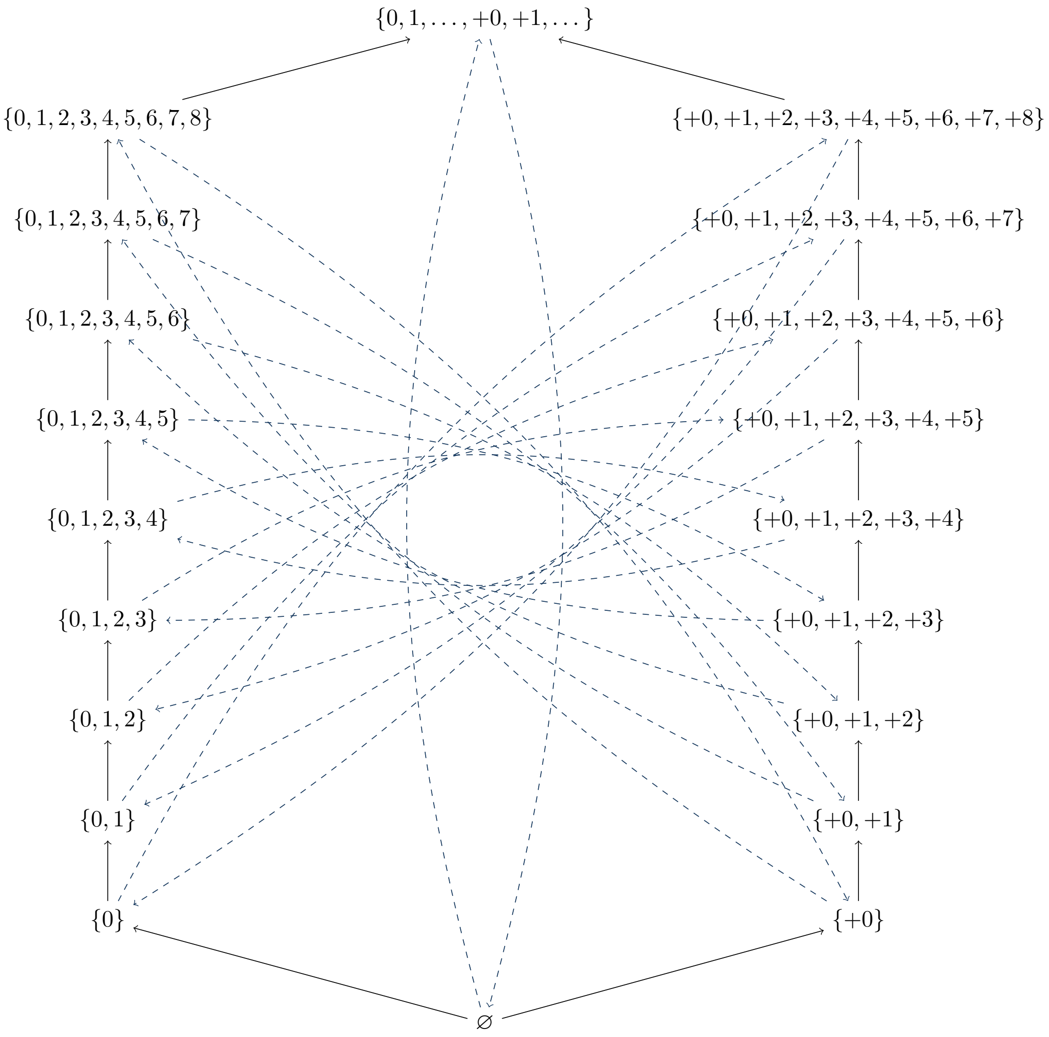


Figure 3. The lattice of types when successful interaction produces a small integer (here less than 9).

We see in this example that the simple fact of being able to discriminate between results, depending on whether they are greater or lesser than 9, allows to have a grasp on all the integers smaller than 9. Indeed, while in the first example, all the integers (and dually, all the offsets) are completely generic, by changing the notion of orthogonality, we are now able to distinguish between all the small integers, by considering which types they are elements of: for instance, 7 is an element of the type $\{1, 2, 3, 4, 5, 6, 7\}$, but not of the type $\{1, 2, 3, 4, 5, 6\}$ and can actually be completely characterized by this property. Not only are small integers individuated purely by looking at their types, but they have some structure. For instance, it is possible to define an order relation over the small integers, considering a number to be smaller than another if the second one is in the biorthogonal of the first (formally defining the relation $n < m$ by $\{n\}^{\perp\perp} \subseteq \{m\}^{\perp\perp}$).

Hence, we see that orthogonality creates a web of types, and the recognition of the singularity of a specific element (here the number 9)[26] causes some other structure to

[26] This singularity could be associated to what, in Section 2, we called the informational content of that element within the set of elements defining its characteristic content.

be immediately visible.

Example 4.3. Another direction that we can follow is to enrich the language, and consider not only that it consists of numbers and offsets, but of any composition thereof as well. In a way, this amounts to internalizing some of the constructions as part of the language itself, such as the composition and the error.

More formally, we take our expressions to be either atomic expressions (integers 0, 1,...; offsets +0, +1,...; and the error $\Box$) or, by induction, any composition of two expressions. Hence, we accept expressions corresponding to chains of computations, such as:

$$\begin{aligned} ((8)(+2))((3)(\Box)) &\rightsquigarrow (+10)((3)(\Box)) \\ &\rightsquigarrow (+10)(\Box) \\ &\rightsquigarrow \Box \end{aligned}$$

with the new rule that computations can happen inside any expression, and that the error $\Box$ propagates itself by deleting whatever it interacts with.

In that case, reusing the orthogonality of Example 4.1, we can see that the orthogonal of an integer is not only an offset but any expression that will be, after computation, reduced to an offset, hence, expressions such as $(+2)(+3)$ but also $(+8)((+6)(+5))$, etc. Indeed, for any integer n, we have the following evaluation process:

$$\begin{aligned} ((+8)((+6)(+5)))(n) &\rightsquigarrow ((+8)(+11))(n) \\ &\rightsquigarrow (+19)(n) \\ &\rightsquigarrow n + 19 \end{aligned}$$

which terminates in an integer.

We see here that the construction of types through orthogonality is capable of handling some kind of compositionality: atomic types can capture common behaviors of expressions which do not share the same composition. For the same reason, the set containing only offsets $\{+0, +1, \dots\}$ is not a type: indeed, there is no way to distinguish, having only access to the results of the computation, between one atomic offset and a computation terminating on an offset. In every situation where $(+2)(+3)$ interacts successfully, so does $+5$ and vice versa.

Let us call **int** the set of integers and **off** the one of offsets. We have that **int** is included but different from $\textbf{int}^{\perp\perp}$ which contains not only numbers, but also every computation that ends on a number.

$$\begin{tikzcd}
\textbf{int} & \textbf{int}^{\perp} = \textbf{off}^{\perp\perp} & \textbf{int}^{\perp\perp} = \textbf{off}^{\perp} \\
\{0,1,2,\dots\} \arrow[r, dashed, bend left] & \{+0,(+0)(+0),(+1)(+3),\dots\} \arrow[r, dashed, bend left] & \{0,(+3)(1),(+0)((2)(+4)),\dots\} \arrow[l, dashed, bend left]
\end{tikzcd}$$

Now, based on that compositional aspects of atomic types such as $\textbf{off}^{\perp\perp}$, we can start to build *derived types*, by introducing, for instance, the arrow connective. Recall that given two types A and B, we define the type

$$A \to B$$

as the set of all terms t such that, applied to any term u of type A, tu is of type B. We can check that it is the orthogonal of a set, and hence a type[27].

For the time being, we have chiefly built types, leaving aside the fact that they type terms. We can now say that a term t has type A (i.e. $t : A$) if it is an element of the type A. In this way, we can see that any integer has, not only the atomic type $\mathbf{int}^{\perp\perp}$, but also the derived type $\mathbf{off}^{\perp\perp} \to \mathbf{int}^{\perp\perp}$, since by interacting with an offset, it results in another integer. Likewise, any offset has the two derived types

$$\mathbf{off}^{\perp\perp} \to \mathbf{off}^{\perp\perp}$$
$$\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}$$

but also more complex ones such as

$$(\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}) \to (\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp})$$

while the error has, for any type A, the type

$$\varnothing \to A.$$

Indeed, it can be successfully applied to nothing (i.e. any member of the empty set), and by doing so produces a (non-existent) element of any type A[28].

The introduction of derived types provides a powerful tool to make explicit the interaction upon which atomic types were built. Based on them, that interaction can be represented as a system of logical dependencies. Significantly, in the case of the very simple connective we have introduced, we can see that it allows for a form of recursion in typing complex expressions. In particular, we can perform so-called type derivations, in the style of natural deduction (Gentzen 1935a,b; Prawitz 1965).

Suppose that we have already established the following three typing judgments:

$$\begin{aligned} +8 &: (\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}) \to (\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}); \\ +6 &: \mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}; \\ 17 &: \mathbf{int}^{\perp\perp}. \end{aligned}$$

These are all judgments about atomic integers and atomic offsets. We can combine them using the *rule of elimination of the arrow*:

$$\frac{t : A \to B \qquad u : A}{tu : B}$$

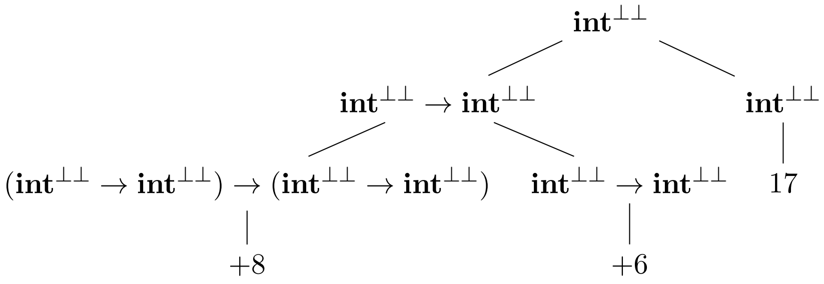
in a derivation typing the composite expression $((+8)(+6))(17)$:

$$\dfrac{\dfrac{+8 : (\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}) \to (\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}) \qquad +6 : \mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}}{(+8)(+6) : \mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}} \qquad 17 : \mathbf{int}^{\perp\perp}}{((+8)(+6))(17) : \mathbf{int}^{\perp\perp}}$$

[27] Consider a set X such that $X^{\perp} = B$, and C such that C is the set of all the results of interactions of an element of A with an element of X. $A \to B$ is the orthogonal of C.

[28] This is a computational interpretation of the logical principle *ex falso quodlibet*: a proof of this principle is an error handler, that produces anything if applied to a failed computation.

Therefore, by performing a type derivation which only depends on judgments on atomic elements and a generic rule of elimination of the arrow (which is justified by the definition of the arrow), we are able to establish how the composite expression interacts (here, as an integer) without having to perform its computation. This can actually be rewritten in a syntax-tree form (from bottom to top):


$\mathbf{int}^{\perp\perp}$
$\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}$
$\mathbf{int}^{\perp\perp}$
$(\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}) \to (\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp})$
$\mathbf{int}^{\perp\perp} \to \mathbf{int}^{\perp\perp}$
17
+8
+6

Typing derivations are thus a way to reduce the typing of complex expression to atomic ones. Typing atomic expressions remain however the toughest part, and there is no general strategy to do so: indeed, in many interesting formal systems, the problem of type-checking—verifying that a term has a given type—is undecidable (it is the case of Girard's System F (Wells 1999)). Nonetheless, in some situations, it is possible to prove so, or to reduce the problem of showing that a term interacts well with all the elements of a type to the problem of showing that it interacts well with finitely many of them.

It follows that, by assuming the compositional aspect of types, not only can we refine the classificatory capacities of types[29], but also capture and represent the structure of type dependencies through derivations emerging from the interaction of terms.

To sum up, within this framework, a type is a set of terms that all interact in the same way with respect to a given set of terms. This construction only depends on the existence of a notion of successful interaction. Examples of this construction abound in the domain of logic and theoretical computer science (see Curien et al. (2010)), and interaction is either a process of algebraic nature (and its success is measured by computing a specific value), or more dynamic, such as in game semantics, where interaction consists of a sequence of moves and is considered successful if one of the two players wins. From this notion a rich algebra of types emerge, that are related in particular through connectives. Depending on the starting interaction, different connectives can be observed. Let us just cite three that are central in Linear Logic (Girard 1987):

- the multiplicative conjunction $\otimes$: an element of type $A \otimes B$ contains both an element of type A and one of type B and each can be used in parallel (or syntagmatically)
- the additive disjunction $\oplus$: an element of type $A \oplus B$ is either an element of type A or an element of type B, but not the two of them at the same time
- the additive conjunction &: an element of type $A \,\&\, B$ can be used both as an element of A and an element of B but not both at the same time.

[29] Which can actually be considered as at the core of classification theory. See Joinet (n.d.).

5. Interaction in Natural Language

By conceiving and constructing types through interaction, we grasp the idea that atomic types are not structureless but already contain traces of the principles of their mutual relationships. The structure of these types either exposes interesting features of the interacting elements (as we saw for small integers) or already contains the seeds of the interaction of types.

To the best of our knowledge, this approach to types through interaction has not yet been applied to the treatment of natural language in a way that can contribute to the intelligibility and development of current NLP methods[30]. The capabilities exhibited by this framework permit to suggest that a proper interpretation of interaction within natural language can help developing current distributional methods in the direction established by the structuralist hypothesis, by addressing the challenges to which the latter is confronted. In particular, the interpretation of paradigms as types built through interaction provides interesting hints on how to deal with the obstacles presented in Section 3.

5.1. *Circularity of paradigm derivation*

The first obstacle we identified was the circularity involved in the derivation of paradigms, given by the circumstance that each paradigm requires other paradigms to be constructed, which require other paradigms in turn, including the one to be derived in the first place. Now, by understanding types as sets which are the orthogonal of other sets, types are conceived as the sets that are stable by the operation of correct interaction. The circularity is, in a way, built in the definition of orthogonality, and the types are exactly the fixed points of this circularity, built together with their dependencies with other types. That A is a type means nothing more (and nothing less) than that there is a certain dependency (captured by a notion of interaction) between the terms in A and other classes of terms, which can, in turn, be constructed as other types, thanks to the action of A (i.e. by being included in their bi-orthogonal).

As already said, all that is needed to put this framework into practice is adequate notions of interaction and successful interaction defined over terms and classes of terms. In the case of the programming languages extending the λ-calculus, the successful interaction governing those principles is taken to be the termination of computational processes (see Riba (2007))[31]. In principle, there is no unique natural way of defining such interaction in the case of natural language. Intuitively, however, interaction can be associated with distributional properties, i.e. two terms interact if they co-occur within linguistic contexts across a given corpus. But we have also suggested that two terms correctly interact by forming a unit of higher level, which requires that the notion

[30] It is worth mentioning here, however, a whole tradition which also applies types defined through interaction to the study of argumentative dialogues: in such a dialogue, each player gives arguments (terms) which interact by creating a dialogue; the dialogue either terminates when one of the two players is convinced by the arguments laid before them (in which case the interaction is successful) or continues *ad libitum*, with no player satisfied with the outcome (in which case the argumentation is unsuccessful). See Fouqueré et al. (2018) for a recent survey of this tradition, which aims at accounting for rhetoric and social interaction between cognitive agents. Without judging the fruitfulness of this approach, it seems clear that it is not concerned with the structure of language as such, which is the object of our inquiry. The relationship between this conversational approach and the one presented here remains to be studied.

[31] The centrality of termination as a criterion for interaction deserves a proper philosophical investigation: indeed, other notions, such as *productivity*—the fact that interaction always produces some information after a finite time—or the property of being *error-free*, could be equally interesting candidates.

of successful interaction take into account at least some paradigmatic properties upon which combinatorial principles could be assessed. This conforms to the circumstance that types are defined through a notion of orthogonality that has been lifted from terms to classes of terms, recovering the idea already hinted at that paradigms contract dependencies with paradigms, and not with terms. Moreover, if the co-occurrence of terms within a certain scope can give a satisfactory idea of linguistic interaction, successful interaction cannot be restricted to the mere presence of terms within a context, since two terms might be able to interact successfully even if they have never been uttered before (or more concretely, even if they do not co-occur within contexts across the given corpus).

From this viewpoint, usual distributional methods, relying on the occurrence of terms, seem insufficient, if not inadequate, to provide the intended notion of successful interaction for natural language. They are not altogether useless, however. Indeed, since, as we have seen, the construction of types begins by considering classes of terms, the fundamental distributional phenomenon can provide relevant classes in a linguistic framework (which is far less controlled than the simple formal setting of integers and offsets of the preceding section). Take, for instance, the phrase `she must know`. Following the classical distributional method, we can look, in a given corpus, for the most frequent words that appear at each position of the context. We can then expect to obtain something similar to the following three classes of terms[32]:

A	B	C
`he`	`could`	`also`
`i`	`did`	`be`
`one`	`may`	`do`
`she`	`might`	`get`
`they`	`must`	`go`
`we`	`should`	`have`
`you`	`would`	`know`
	`′d`	`make`
	`will`	`not`
	`′ll`	`take`

If we call A, B, C those three classes, we can now consider the set $A \times B \times C$ containing all the possible combinations of the terms of those classes, in that order (eg. `he did get`, `they could be`, `we should have`, etc.). Now, the first remarkable fact about this paradigmatic construction is that it provides a generalization of the linguistic data to be analyzed. Because, for all we know, some of the phrases in $A \times B \times C$ (and in fact most, in the general case) might not exist in the corpus, although they constitute perfectly correct expressions of the language under study. As a result, the analysis can be carried beyond the original available data[33]. But more significantly, the

[32] For this toy example, we compute the paradigmatic classes rather naively, using Google Books Ngram Viewer (Michel et al. 2010, https://books.google.com/ngrams), with the following parameters: `en_2019` corpus, from 1900 to 2019, with a smoothing of 3. The use of wildcards permits to recover the most frequent (up to 10) words at a given place, to which it is convenient to systematically add the original word for that place (not needed in the present example). Since this toy example has mostly illustrative character, we disregard the difference in frequency of words within each class, and we order them alphabetically.

[33] Of course, a significant number of incorrect or ungrammatical expressions can also be expected to belong to this set, and we have, in principle, no means to distinguish them from the others. Yet, we can hope that a conservative approach to the extraction of distributional classes (adopting, for instance, longer contexts and high frequency thresholds), a comparison of this set with the expression of the initial corpus and a progressive or iterative filtering of the set based on the statistics of the types to be constructed thereupon can contribute

set $A \times B \times C$ constitutes an idealized linguistic setting (an artificial corpus, as it were, derived from the original one) where types can be constructed in the way presented in the previous section. In particular, we could consider that successful interaction amounts to co-occurrence within the phrases of this idealized corpus. Accordingly, we can say that two sets A and B are orthogonal if all the of terms of A co-occur with the terms of B. In the framework of our idealized corpus, this is of course true by construction. But what is important is that for any subclass of the initial classes, the orthogonals and biorthogonals are stable and can be treated as types. In our simple example, we can restrict the notion of successful interaction to simple concatenation of terms. Since the interaction between terms is not commutative in this case, a given set A will have two orthogonals: its left-orthogonal ${}^{\perp}A$ which contains all the terms with which it interacts well on the left, and its right orthogonal $A^{\perp}$, which is the same on the right. For example, If we take any subclass of the class B defined before, say $\{\texttt{may}, \texttt{might}, \texttt{must}\}$, we have that its left orthogonal coincides with the class A and its right orthogonal is equal to C, both of which become types following our definitions. Moreover, we can consider the (right or left) bi-orthogonal of $\{\texttt{may}, \texttt{might}, \texttt{must}\}$, which is equal to B, and thus also a type. In diagrammatic form, drawing in blue the left-orthogonality and in green the right-orthogonality:

$$\begin{tikzcd}
& \{\texttt{may}, \texttt{might}, \texttt{must}\} \arrow[dl, dashed, bend right] \arrow[dr, dashed, bend left] & \\
A \arrow[r, dashed, bend left] & B \arrow[l, dashed, bend left] \arrow[r, dashed, bend left] & C \arrow[l, dashed, bend left]
\end{tikzcd}$$

In this way, we have constructed three types which are nothing more than the expression of the mutual dependencies that hold between them. Such paradigms can behave like idealized paradigms which could be further used to analyze the initial corpus, by computing statistical properties as linguistic units in their own right, for instance. What is more, their formal construction permits to mobilize the entire type-theoretical apparatus presented in the previous section in such a way that the remaining obstacles concerning paradigm derivation can be addressed in a new perspective.

5.2. *Compositionality*

Another obstacle mentioned in Section 3 was that of the composite organization of linguistic contexts, which could be understood as the problem of establishing syntagmatic relations between paradigmatic units. We suggested that compositional principles capable of supporting hierarchical constructions could provide an adequate solution to this obstacle.

In the type-theoretical framework advanced here, compositionality is related to the different connectives we can build to connect types. If we turn back to our example, we can see that the type B concern principally modal verbs. It is not unreasonable to think that the consideration of other phrases in the corpus would reveal the significant presence of similar types in such a way that a type Mod of modal verbs can be established as a formal unit referring to all those similar types. Moreover, suppose that we are also able to identify in a similar way a type V of verbs. By internalizing the syntagmatic relation in a nod towards linear logic, we can introduce a new type

to reducing the amount of incorrect expressions to a level in which their effect on the regularity of paradigmatic structure derived is marginal.

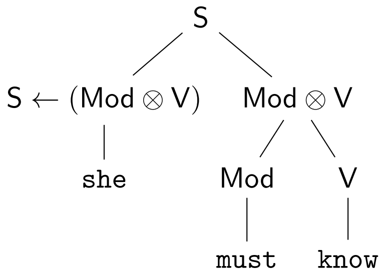
$\mathsf{Mod} \otimes \mathsf{V}$ of composite terms, whose first element is in Mod and second in V. Hence, given the extensions of these types, we can prove that `must know` has such a type:

$$\frac{\texttt{must} : \mathsf{Mod} \qquad \texttt{know} : \mathsf{V}}{\texttt{must know} : \mathsf{Mod} \otimes \mathsf{V}}$$

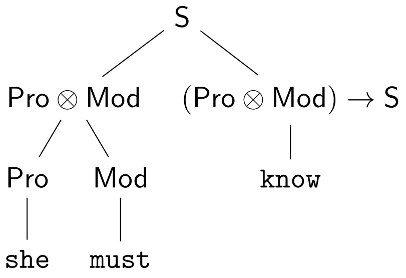
Moreover, `she` is an element of the left-orthogonal of $\mathsf{Mod} \otimes \mathsf{V}$, by which we mean that concatenating `she` to the left of an element of type $\mathsf{Mod} \otimes \mathsf{V}$ generates an element of the corpus. Another way of expressing this idea, which opens the door to an iterative definition, is to state that `she` has type $\mathsf{S} \leftarrow (\mathsf{Mod} \otimes \mathsf{V})$[34]. From which we can draw the derivation tree:

$$\frac{\texttt{she} : \mathsf{S} \leftarrow (\mathsf{Mod} \otimes \mathsf{V}) \qquad \dfrac{\texttt{must} : \mathsf{Mod} \qquad \texttt{know} : \mathsf{V}}{\texttt{must know} : \mathsf{Mod} \otimes \mathsf{V}}}{\texttt{she must know} : \mathsf{S}}$$

This derivation tree can also be presented in a syntax tree form (as in Example 4.3[35]):



Dually, if we start from the other direction (considering that a type Pro of pronouns exists in the corpus, and observing that `know` is orthogonal to $\mathsf{Pro} \otimes \mathsf{Mod}$), we can consider the other syntax tree:



and these two ways of typing the sentence are equally legitimate from the sole analysis of the sentence. It should be nonetheless possible to discriminate between the two variants by resorting, for instance, to statistical properties of these constructions in the original corpus.

It follows that connectives allow to express forms of composition. More can be expressed with other connectives, which allow to express the relationship between types. For instance, in a richer corpus, one can imagine that the type of verb phrases

[34] Denoting S the universal type, and $\leftarrow$ the right-to-left non-commutative version of the connective $\rightarrow$.

[35] Note that we built "syntax trees" in the example, but they are not really trees: indeed, as the orthogonality is commutative, the arrow built in this way has a form of associativity: commutativity at the level of terms creates associativity at the level of types. As this is not the case for natural language, this gives a *a posteriori* justification for the use of trees.

would be exactly the union of the type V of verbs and $\mathsf{Mod} \otimes \mathsf{V}$. Hence, the type of verb phrases can be characterized as

$$\mathsf{V} \oplus (\mathsf{Mod} \otimes \mathsf{V})$$

of either verbs or modals followed by verbs. In the same way, the connective & is a form of intersection and can thus represent the composition of two paradigms consisting of those terms that are part of both.

5.3. *Subtyping and paradigms*

If we take a closer look to our previous example, we can see that it relies not only on the orthogonals of the three classes we presented, but also on our ability, among the third type

$$C = \{\texttt{also}, \texttt{be}, \texttt{do}, \texttt{get}, \texttt{go}, \texttt{have}, \texttt{know}, \texttt{make}, \texttt{not}, \texttt{take}\},$$

to discriminate a specific subset $\mathsf{V} = \{\texttt{be}, \texttt{do}, \texttt{get}, \texttt{go}, \texttt{have}, \texttt{know}, \texttt{make}, \texttt{take}\}$ representing verbs, from $\mathsf{V} = \{\texttt{also}, \texttt{not}\}$ containing adverbs. This problem corresponds precisely to the third obstacle presented in Section 2, where a paradigm like $(\texttt{home}, \texttt{down}, \texttt{in})$ in Figure 1 appeared as the unanalyzed union of smaller ones.

The difficulty resides in the fact that, within the context considered, there is no means to operate the necessary distinction. Yet, from our type-theoretical approach, the intended subset can emerge as a type if other interactions are taken into account. For instance, if we consider, in the way described, the most frequent words coming after `she must also` and after `she must not`, we get the following classes, respectively:

$$\{\texttt{take}, \texttt{be}, \texttt{know}, \texttt{show}, \texttt{have}, \texttt{make}, \texttt{keep}, \texttt{sign}, \texttt{understand}, \texttt{learn}\}$$
$$\{\texttt{be}, \texttt{know}, \texttt{go}, \texttt{have}, \texttt{forget}, \texttt{only}, \texttt{let}, \texttt{do}, \texttt{think}, \texttt{expect}\}$$

So, the specific type V can be thought of as built from other interactions. In this way, such a type will appear associated to the type C through a subtyping relation, akin to the ones presented in the lattices of Figures 2 and 3.

Conclusion

In this paper, we have taken the viewpoint that understanding the recent success of DNN models in NLP requires less to understand the specific technological achievements—regardless of how impressive they are—than to critically reconstruct the image of language that features the properties technological devices mobilize and encourage. The distributional hypothesis, associating the meaning of words with their distribution—the contexts in which they appear—offers such a picture and has been used as a theoretical background to explain the success of the application of DNNs in natural language. We have proposed that this hypothesis should be seen as a weak version of a more sophisticated theoretical stance, rooted in structuralist linguistics, which conceives linguistic units as determined by a dual relationship: paradigmatic and syntagmatic. In particular, we have argued that the distributions have to be understood inside specific paradigms, and not in general, and that the reconstruction of the corresponding paradigmatic system can contribute to drawing structural features underlying the linguistic data under analysis.

However, the structuralist view of language is not free of difficulties. In particular, we have mentioned three main obstacles to a formal implementation of paradigm derivation: the risk of circularity, the need of a hierarchical compositional principle (syntagmatic relation between paradigmatic units) and the necessity of analyzing subparadigms within paradigms (paradigmatic relations between paradigmatic units). We argue that interpreting paradigms as types, as in proof theory, defined through their interaction, can offer fresh perspectives on these problems, by bringing to the fore a notion of interaction. In this way, we can develop an algebraic and logical view of natural language through types, which can eventually meet other research trends such as categorial grammars. Yet, the significance of the framework proposed here lies in that, although algebraic, the analysis remains grounded in a notion of interaction which is derived from the statistics of corpora. As a consequence, statistical and algebraic approaches could coexist within this setting, in a way which integrates also information-theoretic views of language.

Pushing these connections between fields is an exciting conceptual enterprise, but needs to be grounded in more empirical results. The verification (or the falsification) of our theses is not out of reach: the mechanism we presented for building types through statistical analysis of corpora can be, in principle, implemented, and the notion of interaction can thus be refined in order to build meaningful types to be used in real life.

The interactive ideas had a decisive influence in the field of proof-theory and theoretical computer science. We hope that the clear intellectual affinity between these ideas and structural linguistics can be further expanded and developed. Computational linguistics seems to offer a ground for testing the alliance of such ideas.

References

Abzianidze, Lasha. 2016. “Natural Solution to FraCaS Entailment Problems.” In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, Berlin, Germany, Aug., 64–74. Association for Computational Linguistics. https://www.aclweb.org/anthology/S16-2007.

Apostel, L., B. Mandelbrot, and A. Morf. 1957. *Logique, Langage et Théorie de l’Information.* Presses Universitaires de France.

Avraham, Oded, and Yoav Goldberg. 2017. “The Interplay of Semantics and Morphology in Word Embeddings.” In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Valencia, Spain, Apr., 422–426. Association for Computational Linguistics. https://www.aclweb.org/anthology/E17-2067.

Bengio, Yoshua. 2008. “Neural net language models.” *Scholarpedia* 3 (1): 3881.

Blevins, Terra, Omer Levy, and Luke Zettlemoyer. 2018. “Deep RNNs Encode Soft Hierarchical Syntax.” .

Bloomfield, Leonard. 1935. *Language.* London: G. Allen & Unwin, Ltd.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomáš Mikolov. 2016. “Enriching Word Vectors with Subword Information.” *CoRR* abs/1607.04606. http://arxiv.org/abs/1607.04606.

Bradley, Tai-Danae. 2020. “At the Interface of Algebra and Statistics.” *ArXiv* abs/2004.05631.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. “Language Models are Few-Shot Learners.” .

Chater, Nick, Alexander Clark, John A. Goldsmith, and Amy Perfors. 2015. *Empiricism and* *language learnability.* First edition ed. Oxford, United Kingdom: Oxford University Press. OCLC: ocn907131354.

Chen, Tongfei, Yunmo Chen, and Benjamin Van Durme. 2020. "Hierarchical Entity Typing via Multi-level Learning to Rank." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, Jul., 8465–8475. Association for Computational Linguistics. `https://www.aclweb.org/anthology/2020.acl-main.749`.

Choi, Eunsol, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. "Ultra-Fine Entity Typing." In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, Jul., 87–96. Association for Computational Linguistics. `https://www.aclweb.org/anthology/P18-1009`.

Chomsky, Noam. 1953. "Systems of Syntactic Analysis." *The Journal of Symbolic Logic* 18 (3): 242–256. `http://www.jstor.org/stable/2267409`.

Chomsky, Noam. 1957. *Syntactic Structures.* The Hague: Mouton and Co.

Chomsky, Noam. 1969. *Quine's Empirical Assumptions*, 53–68. Dordrecht: Springer Netherlands. `https://doi.org/10.1007/978-94-010-1709-1_5`.

Clark, Kevin, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. "What Does BERT Look At? An Analysis of BERT's Attention." .

Clark, Stephen Hedley, Laura Rimell, Tamara Polajnar, and Jean Maillard. 2016. "The Categorial Framework for Compositional Distributional Semantics." .

Coecke, Bob. 2019. "The Mathematics of Text Structure." .

Coecke, Bob, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. "Mathematical Foundations for a Compositional Distributional Model of Meaning." .

Curien, Pierre-Louis, Hugo Herbelin, Jean-Louis Krivine, and Paul-André Melliès. 2010. *Interactive models of computation and program behaviour.* Société Mathématique de France.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *CoRR* abs/1810.04805. `http://arxiv.org/abs/1810.04805`.

Dinu, Georgiana, Miguel Ballesteros, Avirup Sil, Sam Bowman, Wael Hamza, Anders Sogaard, Tahira Naseem, and Yoav Goldberg, eds. 2018. *Proceedings of the Workshop on the Relevance of Linguistic Structure in Neural Architectures for NLP*, Melbourne, Australia, Jul. Association for Computational Linguistics. `https://www.aclweb.org/anthology/W18-2900`.

Ducrot, Oswald. 1973. *Le structuralisme en linguistique.* Paris: Éditions du Seuil.

Enguehard, Émile, Yoav Goldberg, and Tal Linzen. 2017. "Exploring the Syntactic Abilities of RNNs with Multi-task Learning." In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 3–14.

Firth, John Rupert. 1957. "A synopsis of linguistic theory 1930–1955." In *Studies in Linguistic Analysis*, 1–32. Oxford: Blackwell.

Fouqueré, Christophe, Alain Lecomte, Myriam Quatrini, Pierre Livet, and Samuel Tronçon. 2018. *Mathématique du dialogue: sens et interaction.* Hermann.

Gastaldi, Juan Luis. 2020. "Why Can Computers Understand Natural Language?" *Philosophy & Technology* `https://doi.org/10.1007/s13347-020-00393-9`.

Gentzen, Gerhard. 1935a. "Untersuchungen über das logische Schließen. II." *Mathematische Zeitschrift* 39: 176–210.

Gentzen, Gerhard. 1935b. "Untersuchungen über das logische Schließen. II." *Mathematische Zeitschrift* 39: 405–431.

Girard, Jean-Yves. 1987. "Linear logic." *Theoretical Computer Science* 50 (1): 1 – 101.

Girard, Jean-Yves. 1989. *Towards a Geometry of interaction*, Vol. 92 of *Contermporary Mathematics*, 69–108. AMS.

Girard, Jean-Yves. 2001. "Locus Solum: From the rules of logic to the logic of rules." *Mathematical Structures in Computer Science* 11 (3): 301–506.

Goldberg, Yoav. 2019. "Assessing BERT's Syntactic Abilities." .

Groote, Philippe De. 1995. *The Curry-Howard Isomorphism.* Academia.

Harris, Zellig. 1960. *Structural linguistics.* Chicago: University of Chicago Press.

Harris, Zellig. 1970a. "Distributional Structure." In *Papers in Structural and Transformational Linguistics*, 775–794. Dordrecht: Springer.

Harris, Zellig S. 1970b. *Computable Syntactic Analysis: The 1959 Computer Sentence-Analyzer*, 253–277. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-017-6059-1_16.

Harris, Zellig S. 1970c. *Morpheme Boundaries within Words: Report on a Computer Test*, 68–77. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-017-6059-1_3.

Hewitt, John, and Christopher D. Manning. 2019. "A Structural Probe for Finding Syntax in Word Representations." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, Jun., 4129–4138. Association for Computational Linguistics.

Hjelmslev, Louis. 1953. *Prolegomena to a Theory of Language.* Baltimore: Wawerly Press.

Hjelmslev, Louis. 1971. *La structure fondamentale du langage*, 177–231. Paris: Éditions de Minuit.

Hjelmslev, Louis. 1975. *Résumé of a Theory of Language.* Travaux du Cercle linguistique de Copenhague 16. Copenhagen: Nordisk Sprog-og Kulturforlag.

Jakobson, Roman. 1967. *Preliminaries to speech analysis: the distinctive features and their correlates.* Cambridge, Mass: M.I.T. Press.

Jakobson, Roman. 2001. *Roman Jakobson: Selected Writings.* Berlin Berlin: Mouton De Gruyter.

Joinet, Jean-Baptiste. n.d. "Collusions and agonal quotients : generalizing equivalence relations and definitions by abstraction." *Review of Symbolic Logic* (under review).

Kell, Stephen. 2014. "In Search of Types." In *Proceedings of the 2014 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software*, Onward! 2014, New York, NY, USA, 227–241. Association for Computing Machinery. https://doi.org/10.1145/2661136.2661154.

Krishnamurthy, Jayant, Pradeep Dasigi, and Matt Gardner. 2017. "Neural Semantic Parsing with Type Constraints for Semi-Structured Tables." In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, Sep., 1516–1526. Association for Computational Linguistics. https://www.aclweb.org/anthology/D17-1160.

Krivine, Jean-Louis. 2010. "Realizability algebras: A program to well order R." *Logical Methods in Computer Science* 7 (3).

Lambek, Joachim. 1958. "The Mathematics of Sentence Structure." *The American Mathematical Monthly* 65 (3): 154–170.

Landauer, Thomas K., Danielle S. McNamara, Simon Dennis, and Walter Kintsch, eds. 2007. *Handbook of Latent Semantic Analysis.* Mahwah, New Jersey, USA: Lawrence Erlbaum Associates.

Lenci, Alessandro. 2008. "Distributional semantics in linguistic and cognitive research." *From context to meaning: distributional models of the lexicon in linguistics and cognitive science, Italian Journal of Linguistics* 1 (20): 1–31.

Lenci, Alessandro. 2018. "Distributional Models of Word Meaning." *Annual Review of Linguistics* 4 (1): 151–171.

Levy, Omer, and Yoav Goldberg. 2014a. "Linguistic Regularities in Sparse and Explicit Word Representations." In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*, 171–180.

Levy, Omer, and Yoav Goldberg. 2014b. "Neural Word Embedding As Implicit Matrix Factorization." In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, Cambridge, MA, USA, 2177–2185. MIT Press.

Levy, Omer, Yoav Goldberg, and Ido Dagan. 2015. "Improving Distributional Similarity with Lessons Learned from Word Embeddings." *TACL* 3: 211–225.

Lin, Ying, and Heng Ji. 2019. "An Attentive Fine-Grained Entity Typing Model with Latent Type Representation." In *Proceedings of the 2019 Conference on Empirical Methods in*

Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, Nov., 6197–6202. Association for Computational Linguistics. `https://www.aclweb.org/anthology/D19-1641`.

Linzen, Tal, Emmanuel Dupoux, and Yoav Goldberg. 2016. "Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies." .

MacWhinney, Brian, ed. 1999. *The emergence of language.* Carnegie Mellon symposia on cognition. Mahwah, NJ: Lawrence Erlbaum Associates.

Maniglier, Patrice. 2006. *La vie énigmatique des signes.* Paris: Léo Scheer.

Manning, Christopher D., Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. "Emergent linguistic structure in artificial neural networks trained by self-supervision." *Proceedings of the National Academy of Sciences* `https://www.pnas.org/content/early/2020/06/02/1907367117`.

Martini, Simone. 2016. "Several Types of Types in Programming Languages." In *History and Philosophy of Computing*, edited by Fabio Gadducci and Mirko Tavosanis, Cham, 216–227. Springer International Publishing.

McEnery, Anthony M., and Anita Wilson. 2001. *Corpus Linguistics: An Introduction.* Edinburgh: Edinburgh University Press.

McGee Wood, Mary. 1993. *Categorial grammars.* London: Routledge.

Michel, Jean-Baptiste, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, et al. 2010. "Quantitative Analysis of Culture Using Millions of Digitized Books." *Science* `http://www.sciencemag.org/content/331/6014/176.full`.

Mikolov, Tomáš, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." *CoRR* abs/1301.3781.

Minsky, Marvin L. 1991. "Logical Versus Analogical or Symbolic Versus Connectionist or Neat Versus Scruffy." *AI Magazine* 12 (2): 34. `https://www.aaai.org/ojs/index.php/aimagazine/article/view/894`.

Miquel, Alexandre. 2020. "Implicative algebras: a new foundation for realizability and forcing." *Mathematical Structures in Computer Science* 30 (5): 458–510. `http://dx.doi.org/10.1017/S0960129520000079`.

Moot, Richard, and Christian Retoré. 2012. *The Logic of Categorial Grammars: A Deductive Account of Natural Language Syntax and Semantics.* 1st ed., Lecture Notes in Computer Science 6850. Springer-Verlag Berlin Heidelberg.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. "Deep contextualized word representations." *CoRR* abs/1802.05365. `http://arxiv.org/abs/1802.05365`.

Pilehvar, Mohammad Taher, and Jose Camacho-Collados. 2020. "Embeddings in Natural Language Processing. Theory and Advances in Vector Representation of Meaning." Draft, `http://josecamachocollados.com/book_embNLP_draft.pdf`.

Prawitz, Dag. 1965. *Natural Deduction: A Proof-Theoretical Study.*

Quine, W. V. 2013. *Word and object.* Cambridge, Mass: MIT Press.

Radford, Alec. 2018. "Improving Language Understanding by Generative Pre-Training." .

Raiman, Jonathan, and Olivier Raiman. 2018. "DeepType: Multilingual Entity Linking by Neural Type System Evolution." .

Riba, Colin. 2007. "Strong Normalization as Safe Interaction." In *LiCS '2007*, 13–22.

Sahlgren, Magnus. 2006. "The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces." PhD diss., Stockholm University, Stockholm, Sweden.

Sahlgren, Magnus. 2008. "The Distributional Hypothesis." *Special issue of the Italian Journal of Linguistics* 1 (20): 33–53.

de Saussure, Ferdinand. 1959. *Course in General Linguistics.* New York: McGraw-Hill. Translated by Wade Baskin.

Schnabel, Tobias, Igor Labutov, David M. Mimno, and Thorsten Joachims. 2015. "Evaluation methods for unsupervised word embeddings." In *Proceedings of the 2015 Conference* *on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 298–307.

Schütze, Hinrich. 1992. “Dimensions of Meaning.” In *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, Supercomputing ’92, Los Alamitos, CA, USA, 787–796. IEEE Computer Society Press.

Schütze, Hinrich. 1993. “Word Space.” In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, San Francisco, CA, USA, 895–902. Morgan Kaufmann Publishers Inc.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. “Neural Machine Translation of Rare Words with Subword Units.” In *Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers)*, Berlin, Germany, Aug., 1715–1725. ACL.

Shannon, Claude E. 1948. “A mathematical theory of communication.” *Bell Syst. Tech. J.* 27 (3): 379–423.

Turney, Peter D., and Patrick Pantel. 2010. “From Frequency to Meaning: Vector Space Models of Semantics.” *CoRR* abs/1003.1141.

Wells, J.B. 1999. “Typability and type checking in System F are equivalent and undecidable.” *Annals of Pure and Applied Logic* 98 (1): 111 – 156.

Wijnholds, Gijs, and Mehrnoosh Sadrzadeh. 2019. “A Type-Driven Vector Semantics for Ellipsis with Anaphora Using Lambek Calculus with Limited Contraction.” *J. of Logic, Lang. and Inf.* 28 (2): 331–358. `https://doi.org/10.1007/s10849-019-09293-4`.