

Pattern Lattice as a Model for Linguistic Knowledge and Performance*

Kow Kuroda

National Institute of Information and Communications Technology (NICT), Japan
kuroda@nict.go.jp

Abstract. This paper outlines a theoretical model, called the Pattern Lattice Model (PLM) of human linguistic knowledge and performance, and presents a simple implementation of this model. Any expressions found in a natural language L are structured in some ways, and linguists are willing to assume that those expressions are the products of what they call the *grammar* of L . In contrast, the PLM embodies a “radically memory-based” view of L , and provides a viable alternative to the traditional “grammar-based” model of L . The PLM is also expected to lay the theoretical foundations for the so-called “usage-based model” of language, which lacks solid foundations.

Keywords: pattern lattice model, radically memory-based model, radically usage-based model, vast memory thesis

1 Why a theory of pattern lattice?

1.1 Two views of language

A proposed theory of pattern lattice (PL) was conceived and developed to implement a view of linguistic performances with the following characteristics:¹

- (1) **Memory-based model of language:** In actual performances, people do not “generate” sentences. Instead, they “blend” some of the sentences that they “remember,” irrespective of whether they speak/write (in production) or hear/read (in comprehension). The sentences chosen to be blended have “partial matches” to a target meaning (in production) or source sentence (in comprehension). The sentences chosen often undergo “edits” while blending.

Some terms such as “blend”, “remember,” “partial matches,” and “edits” in this statement are used metaphorically; however, it is intentional. We will eventually consider what they actually mean. However, why is such a model required? In short, it is required to overcome the dominance of the “grammar-based model” of language in the sense described below. Nonetheless, some clarifications would be helpful here.

First, the view embodied in (1) can be interpreted as the generalization of a new view of “rich phonology” by Port (2007). Its essential property lies in its rejection of the abstract representations assumed in “autonomous” phonology. Second, and more importantly, the model of language outlined in (1) lays the foundations for the so-called “usage-based model” of language (Langacker, 1988; Bybee, 2001), which lacks solid theoretical foundations.² The model was proposed as an alternative to the well-accepted, nearly “standard” model of language roughly characterized in (2):

- (2) **Grammar-based model of language:** People produce a given sentence by “combining” a finite set of “elements” (e.g., “words” or “lexical items”) under an algebraic system called “grammar.”

The view of linguistic competence and performance outlined in (2) is not only traditional in linguistics but also well accepted in many fields of cognitive science related to it. It is championed by Generative

* The earlier drafts of this paper benefited from the comments of Masato Yoshikawa (Keio University) and Kimihiko Kuromiya (Osaka Gakuin University), and the final version from anonymous reviewers. The remaining errors, however, are mine. This work was supported in part by Grant-in-Aid for Scientific Research (B) (19330156) from the Japan Society for the Promotion of Science.

Copyright 2009 by Kow Kuroda

¹ The pioneering advocates of the memory-based view of language include Daelemans and van den Bosch (2005). Despite the importance of their work, I do not consider the line of research in this paper due to space limitation.

² Although one could argue that Tomasello (2003) can be a viable candidate for this, I think that it is rather dubious. His framework depends crucially on the unexplained mechanism of “schematization,” whose details, as §1.4.2 shows, are far from obvious.

Grammar (Chomsky, 1965) and its variants.³ In the following, I argue that the memory-based model needs to be preferred to the grammar-based model.

1.2 Ground for the memory-based view

Port (2007) recently advocated a version of a strongly memory-based view of phonology, which he calls “rich phonology.” The essential points made by Port are as follows:

- (3) **Vast Memory Thesis (VMT):** Many aspects of human linguistic performance can be accounted for only by assuming that they are based on vast episodic/exemplar memories with complete details (rather than on a series of computations over abstract representations like “rules” and “schemas”).

Admittedly, such a view was not traditional either in linguistics or in related fields in cognitive science.⁴ However, the situation is changing. For example, Hawkins and Blakeslee (2004) recently proposed, under the name of *Memory-Prediction Framework*, a model of human intelligence in general with a similar concern, and it is gaining popularity.⁵ Even in linguistics, evidence is accumulating that the grammar-based model cannot completely account for the important properties of language such as (i) more importance of noncompositional semantics of collocations than compositional semantics of isolated words,⁶ (ii) existence of endless variations, and (iii) bounded productivity/creativity under conservativeness. Let me explain them briefly in turn.

Research in machine translation (MT), or more exactly its failure, for over 40 years strongly suggests that the compositionality of expressions that grammar guarantees is an illusion. If the semantics of natural language were as much compositional as expected, the rule-based MT systems would have been enough, and therefore, we would not have needed any statistical MT system to replace them. The systematic failure of the rule-based MT systems suggests that what makes sentences of a given natural language meaningful is a large collection of collocations and conventional, prefabricated ways of expressing ideas, as revealed through research in corpus linguistics such as Sinclair (1991). Noncompositionality resides in and brews inside them. This simply suggests that the grammar-based model is incomplete.

I shall focus on bounded productivity/creativity under conservativeness. It can be shown that the realization space of natural language expressions is highly sparse, and arguably, any natural language consists of far less variations in terms of expression types than its grammar allows. In other words, natural language is conservative in its degree of allowance for truly new expressions. This is the property that Wray (2002) calls the “formulaicity” of natural language. On the other hand, it is easy to see that a natural language allows endless variations. Apparently, there seem to be some basic patterns that are finite in number. However, there are many variations of these basic patterns. Let us call them first order variations. There are also many variations of those first order variations as well. We may call them second order variations. Furthermore, there are many variations of the second order variations. This ramifications seems to be endless, implying that we have variations of n th order with n being an indefinite integer. Oddly enough, though, the variations seem to be impoverished compared to the outputs of well-constrained (lexicalized) grammars. Thus, the combination of the conservativeness and allowance for endless variations defines what I call “bounded productivity/creativity under conservativeness” in natural language production, by which This implies that, on the one hand, productivity of natural language is constrained under strong tendency for conservativeness, and on the other hand, there are endless variations of expressions.

³ Even the practitioners of the so-called “cognitive linguistics” cannot escape from this grammar-based view because they merely adopt an alternative image of grammar. There is no difference between generative linguists and cognitive linguists in that both do assume something called “grammar.” Notable exceptions would be *Emergent Grammar* (Hopper, 1987) and *Radical Construction Grammar* (Croft, 2000).

⁴ Notable exception is the phenetics-oriented phonology advocated by J. Pierrehumbert and her colleagues.

⁵ Hawkins’ framework has developed into the model of *Hierarchical Temporal Memory (HTM)*.

⁶ One of the reviews pointed out that my argument against the putative compositionality of linguistic semantics can be blurred if one relies on the distinction of “interpretation” from “semantics” in the way of Barwise and Perry (1983). I understand the point, but I cannot accept it for the following reasons. First of all, the tenet of the suggested argument is to protect linguistic “competence” from “performance.” Note that the relation of semantics to interpretation in the relevant sense is an analogue of the relation of competence to performance. I am afraid that the hypothetical distinction between semantics and interpretation is as much illusory as the distinction between competence and performance. The theoretical position that I take is operational minimalism (aka Occam’s Razor). In this position, linguistic performance (including semantic interpretation) is the only (observable) phenomena that deserves a scientific explanation. Crucially, the less an explanation depends on external assumptions like linguistic competence (including semantics), the better it is. In short, I am trying to put aside the notion of semantics by preferring the notion of interpretation over it. Whether successful or not, I believe that I am consistent in that I reject the notion of semantics in the same way that I reject the notion of competence.

Thus, it is a paradox that natural language has endless variations even if it has only a limited generativity as far as the grammar-based view is adopted. Under the memory-based view, it is enough to assume that linguistic production is memory-based rather than grammar-based to account for the seemingly limitless ramification of variations. The grammar-based view, in contrast, faces a series of fundamental difficulties in its attempt to explain them.

Thus, I conclude that the memory-based view is more preferable than the grammar-based view if serious application is considered. However, the view, if understood in the strongest form, also meets a set of conceptual problems. I shall now focus on them.

1.3 Prima facie problems with VMT

The VMT in (3) allows us, at least theoretically, to assume that people remember the utterances that they hear (and the sentences that they read) in real-life, and the sentences that they compose in their minds. Some readers may wonder if this assumption is valid. Clearly, it needs to be examined.

The first and most obvious challenge to the VMT is the issue of memory limit. I am aware that the VMT runs totally counter to many people's strong intuition that human memory is unreliable. This necessitates a defense for the VMT. For this, we rely on two-fold distinctions; the distinction between *memorizing* (= "storage") and *remembering* (= "recall"), and the distinction between *explicit* and *implicit* memories.

It is crucial to understand that the storage of "records" in the memory (= *memorizing*) and the retrieval of records stored in the memory (= *remembering*) are conceptually different.⁷ Note that remembering presupposes storage but not vice versa. It is possible, at least theoretically, to imagine an obviously useless memory system in which everything is stored but nothing can be retrieved. This explains the following interesting property of a memory system: Suppose that you made an observation that a memory system *m* does not remember *r*. You cannot tell, based on this, whether (case 1) some record *r* is not stored in *m* or whether (case 2) *r* is made inaccessible for some reason.⁸ It follows from this that people's intuition related to their own memories needs not always be a reliable evidence for the rejection of the VMT: case 2 can always be true, at least theoretically.

Remembering has an important property related to the last point: *explicit* and *implicit* memories are most likely to be two different types of remembering/recall. The records in explicit memory can be remembered along with the sense of remembering, but those in implicit memory are not accompanied by such a sense. Arguably, implicit memory is at the base of *implicit learning* performed by H.M., a well-studied patient of anterograde amnesia (Milner *et al.*, 1968). Humans subconsciously memorize and learn about many things. This forms the second ground for the VMT. Under the distinction, I stress that a large portion of linguistic memory is implicit memory. This property, I have to admit, makes the VMT not easy to falsify; however, on the other hand, it makes the VMT compatible with the apparently contradicting facts of memory distortion known in the literature.

With respect to the memory limit, is it impossible for humans to have a virtually unlimited amount of memory? This does not seem very unlikely under extraordinary memory performances by exceptional figures such as Solomon Shereshevsky (Luria, 1987), and famous patients with the Savant syndrome such as Kim Peek and Daniel Tammet.⁹ These figures are exceptional, but the exact nature of their exceptionality is far from well understood.¹⁰

1.4 True problems with VMT

I, for one, acknowledged and accepted the VMT outlined in (3) with seriousness, but soon realized that it created "new problems" that call for solutions and that did not exist in the grammar-based model of language. This is why I developed a theory of PL to be illustrated in this work. Let me be more specific.

⁷ It would be helpful to note that the term "memory" is biased for the sense of memorizing; however, memory, as a system, does not work without remembering, and that it would be wrong to simply assume that memorizing and remembering are separately implemented. Such a classification is valid on digital computers, but it would not be so on human memory systems.

⁸ This also implies that there are at least two senses of *forgetting* or *losing memory*.

⁹ One of the reviewers complained that these pieces of evidence are just anecdotes that do not conform to a research paper. I am aware of it, but even now, the scientific description of vast memories is sparse. The last option I had was to allude to a few facts of anecdote status.

¹⁰ McGaugh (2003) makes an interesting point regarding this. He suggests that forgetting or the inability to remember one's own experiences is an adaptive behavior rather than "failure." This is not a traditional view, but if we adopt it, it is conceivable, if not theoretically necessary, to suppose that what is damaged in such exceptional people is their failure to acquire the ability to forget, or rather, ignore the irrelevant details of their memory, or to forget the potentially overwhelming details of daily life rather than the acquisition of the ability to remember them.

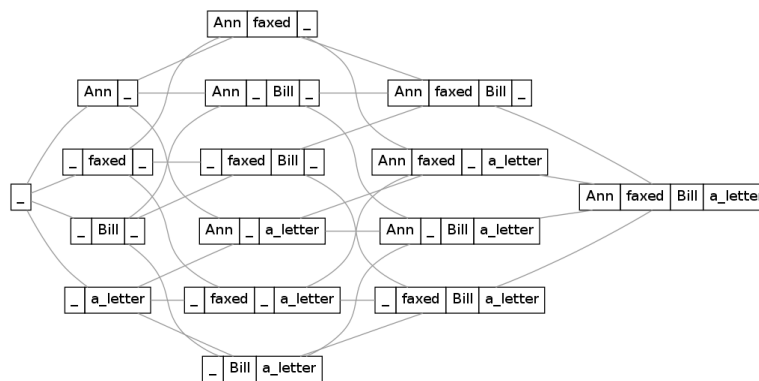


Figure 1: Simplified PL for (5): the top is [-], leftmost, and the bottom, [Ann, faxed, Bill, a letter], rightmost

1.4.1 Two issues with the VMT As far as I can see, any memory-based model of language (or cognitive activities in general) needs to resolve the following crucial issues:

- (4) a. **Issue of realistic storage:** How are sentences stored/represented in (vast) memory?
- b. **Issue of realistic retrieval:** How are stored sentences retrieved/accessed from vast memory and put to use in performances?

We need to admit that there is no simple answer to either of them. In particular, the VMT faces difficulties with respect to the important aspects of human memory performances: people can and do remember many things in a very short time (perhaps on the scale of (milli)seconds). The VMT runs counter to the fact, in the sense that remembering something in a vast memory is like “finding a needle in a huge hay stack.” Thus, the real challenge for the VMT is not the issue of memory limit, but the issue of unexplained efficiency with which records are retrieved or recalled. We can conduct a new research on the memory-based model of linguistic performance only after admitting that it creates a serious implausibility of the VMT.

1.4.2 “Schemas” as indices What makes human memories so efficient? There seems to be no obvious answer, but there surely are certain effective “tricks” in the mechanism of remembering rather than of memorizing. To understand what they can be, we begin by positing that high performances in remembering are realized by the **highly efficient indexing of all instances in memory**, and then, developed a PLM to implement this idea. I am not fully convinced whether this is the right way to go, but nobody knows the right way, since the VMT created a new research field and agenda in a somewhat unexpected way.

At first, efficient remembering by humans was nothing but a puzzle. However, it would be enough to revise the role and notion of the “schemas” of human cognition (Arbib *et al.*, 1987). Arguably, schemas play an important role in many theories of human cognition but their role cannot be the same as in the VMT. The proposed solution is to assume that schemas are best reinterpreted as the “indices” of instances at varying degrees of granularity/specificity. It would guarantee the highly efficient remembering in humans.

1.4.3 Alternative raison-d’être for grammar Under this new definition of schemas as indices to instances stored in a vast memory, it would be crucially helpful to note that the model of language to be proposed below does not dispense with grammar per se: rather, the new model redefines its role. The alternative characterization is that **grammar is a management system for a vast memory of utterance instances**. For this reason, grammar is still a sine-qua-non, but for a different reason. The difference from the traditional grammar-based view is that grammar is no longer a “generative” system, simply because it does not need to be so. It is one of my aims in this paper to show that this is a viable research orientation.

To be precise in the modeling, however, it would be adequate to separate the production and reception of language methodologically. The separation is relevant to the issue under discussion because the former is more difficult than the latter for models compatible with the VMT. This is solely because the well-controlled generation of “new” utterances is more difficult to implement in memory-based systems. By new utterances, I mean undefined combinations of words/phrases that did not exist in the memory. In contrast, the grammar-based models of language are free from such problems. What they suffer from is the overgeneration of unacceptable or unnatural expressions. I argue, however, that this is not so serious a problem as it stands, and that the most serious problem with the VMT lies elsewhere. This is because it

can be solved if we can show that new expressions are constructed by “editing” pre-existing expressions in appropriate ways. For space limitation, this paper does not provide a detailed description but lets it suffice to claim that the unification of “superlexical patterns” in the sense defined in §2.3.1 can model it.

1.5 Partial solution with PL

Suppose that we accepted a version of the memory-based model such as (1) under the VMT in (3). Does this solve the problems and mysteries of human linguistic performances as suggested by Port? The answer is both yes and no: rather, it is not the end but the beginning of a new research. In this work, however, I do not attempt to take the above issue in the general domain. It is the most difficult approach, which would be suitable for ambitious attempts like Hawkins’ HTM. My goals are limited. I try to model human knowledge of language under the VMT. The basic idea is that the human knowledge of collocational units can be represented in terms of a structure to which I will refer as “pattern lattice” (PL), that is, a complete lattice in which all the instances and “patterns” of a language are nodes above the bottom.

A caveat and an excuse: This paper presents a theory of PLs in an extreme form to allow for absurd properties: for example, the entire PL for all the expressions that a speaker knows can be unrealistically large. However, I do so to make as clear as possible the strengths and weaknesses of the theory and facilitate further refinements, because I am sure that the theory is far from complete and satisfactory. This paper is strongly theoretically oriented. I regret not being able to present as many examples as required to make it convincing.

2 Implementing PL

2.1 Current implementation

The Formal Concept Analysis (FCA) (Ganter *et al.*, 2005) was employed to implement the theory of pattern lattice outlined above. The result is made available at (<http://www.kotonoba.net/rubyfca/pattern/>) under the name of pattern lattice builder (PLB).¹¹ For space limitation, the introduction to FCA is omitted in this paper.

2.2 Exposition with simple examples

Even if we strongly rely on the lattice theory, a field of mathematics, it would not be a good idea to begin the exposition with mathematical definitions. Let me begin with simple, concrete examples instead.

2.2.1 Constructing a lattice Presented in Figure 1 is the PL of (5) created by PLB.

(5) Ann faxed Bill a letter.

The structure in (5) is constructed in the following manner:

(6) The pattern lattice for input e is constructed through the following steps:

Step 1: Segment e into an array of segments of desired sizes. Each result is called the “segmentation” of e . Every segmentation is an array of “constants.”

Step 2: Choose one of the possible segmentations.

Step 3: Given an array of segments a chosen, replace all segments in a by variables (denoted by $_$) recursively. Each replacement generates an array of constants and variables, yielding a powerset of arrays. The elements of the powerset are called “patterns” derived from e if they contain at least one variable.

Step 4: (Optional simplification) Reduce consecutive variables into one.

Step 5: Construct an ordered set over the powerset under the “instance-of” relation defined in the following: given a pair of segment arrays (or patterns) a_i and a_j , a_i is an instance of a_j if and only if the k th segment of a_i is i) equal to the k th segment of a_j or ii) k th segment of a_j is $_$.

Step 6: Interpret the ordered set as a lattice under the instance-of relation by letting the original segmentation of e be the “bottom” and the array consisting of variables only be the “top.”¹²

¹¹ This system was implemented by Yoichiro Hasebe (Doshisha University) when he was at NICT in 2008.

¹² This order can be reversed so that the lattice is constructed under the “part-of” relation instead of the “instance-of” relation. This is a property of PL and is not expected to hold generally about lattices.

- (7) Products of this procedure are called “pattern lattices” (PLs).
 (8) A model of language that implements (a version of) the theory specified in (6) is referred to as Pattern Lattice Model (PLM).

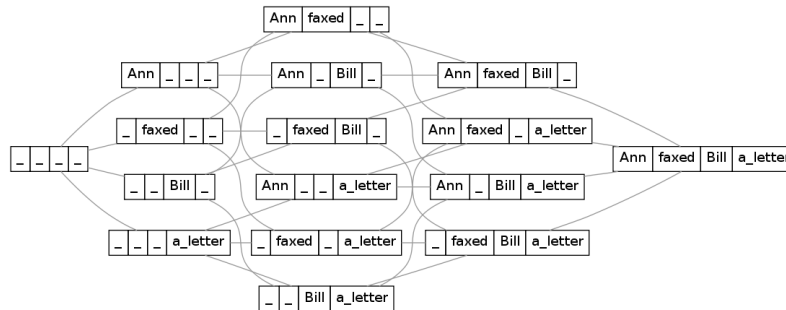


Figure 2: Full PL for (5): the top, at leftmost, is $[-, -, -, -]$, and the bottom, at rightmost, is $[Ann, faxed, Bill, a\ letter]$

Let me explain the procedure step by step. Starting with the example (5), Step 1 produces a set of segmentations like $\{ [Ann, faxed, Bill, a\ letter], [Ann, fax, -ed, Bill, a, letter], [Ann\ faxed, Bill\ a\ letter], \dots \}$. Suppose that we choose $[Ann, faxed, Bill, a\ letter]$ in Step 2.¹³ This choice leads us to the production with either of the following powersets:

- (9) without variable simplification in Step 4: {
 a. $[Ann, faxed, Bill, a\ letter]$,
 b. $[-, faxed, Bill, a\ letter], [Ann, -, Bill, a\ letter], [Ann, faxed, -, a\ letter], [Ann, faxed, Bill, -]$,
 c. $[Ann, faxed, -, -], [Ann, -, Bill, -], [Ann, -, -, a\ letter], [-, faxed, Bill, -], [-, faxed, -, a\ letter], [-, -, Bill, a\ letter]$,
 d. $[Ann, -, -, -], [-, faxed, -, -], [-, -, Bill, -], [-, -, -, a\ letter]$,
 e. $[-, -, -, -]$ }
 (10) with variable simplification in Step 4: {
 a. $[Ann, faxed, Bill, a\ letter]$,
 b. $[-, faxed, Bill, a\ letter], [Ann, -, Bill, a\ letter], [Ann, faxed, -, a\ letter], [Ann, faxed, Bill, -]$,
 c. $[Ann, faxed, -], [Ann, -, Bill, -], [Ann, -, a\ letter], [-, faxed, Bill, -], [-, faxed, -, a\ letter], [-, Bill, a\ letter]$,
 d. $[Ann, -], [-, faxed, -], [-, Bill, -], [-, a\ letter]$,
 e. $[-]$ }

The Hasse diagrams in Figures 1 and 2 correspond to the ordered sets in (10) and (9), respectively.

2.2.2 Note on variable introduction Strictly speaking, the variables denoted by “-” are not syntactic variables; rather, they are the “memory-traces” of certain constants in the sense of Hintzman (1986). We assume them encode semantic and phonological features. We will also focus on this in §2.5.2. If the contents of -’s are tracked and semantically classified in terms of features, the constituency effects to be mentioned in §2.3.4 will automatically follow, although this paper does not explain it.

2.2.3 No (need for) syntactic categories or PoS labels The mechanism of constant-replacement embodies automatic abstraction that dispenses with the assumption of Part-of-Speech specification like N and V or syntactic categories such as NP and VP. Irrespective of the labels attached, variables always correspond to the sets of (series of) constants and capture certain distributional similarities to varying degrees.

2.2.4 No (need for) “ultimate” elements In the proposed model, I reject the traditional view of the hierarchy of linguistically relevant structures as arising from the recursive composition from putative “ultimate” elements. Instead, the PLM assumes that sequences are formed according to stochastic principles such that segmentations are optimally organized and pose the least stress on internal representations.

¹³ No choice of one list over others is justified in the current model. For this, we need a segmentation mechanism that works in an unsupervised fashion. I will focus on this in §2.5.1.

2.2.5 No (need for) a “fixed” lexicon As stated, a choice from segmentations in Step 2 is arbitrary. This implies that optimal segmentation cannot be determined before having the final lattice, on the one hand, and that there need not be a fixed “lexicon,” on the other. This is an intended design choice.¹⁴

2.3 Meaning construction under PL

Table 1: Superimposition of **p1**, ..., **p4** into **p0**

p0	Ann	faxed	Bill	a letter
p1	—	faxed	Bill	a letter
p2	Ann	—	Bill	a letter
p3	Ann	faxed	—	a letter
p4	Ann	faxed	Bill	—

Table 2: Superimposition of **q1**, ..., **q3** into **p2**

p0	Ann	—	Bill	a letter
q1	Ann	—	Bill	—
q2	Ann	—	—	a letter
p3	—	—	Bill	a letter

2.3.1 Basics Let us first introduce a useful terminology. We call patterns containing one and only one constant “lexical patterns.” We call patterns containing more than one constants “superlexical patterns.” In the example above, [Ann, —], [Ann, —, —, —], [—, fax, —], [—, fax, —, —], [—, Bill, —], [—, —, Bill, —], [—, a letter], and [—, —, —, a letter] are lexical patterns. [Ann, faxed, —], [Ann, faxed, —, —], [—, fax, Bill, —], [—, faxed, —, a letter], [Ann, —, a letter], and [Ann, —, —, a letter] are superlexical patterns. Patterns like [—] and [—, —, —, —] are called “null pattern,” or “top pattern.” The rank of patterns is defined as follows: In a PL constructed for an array with k -segments, the number of constants in a pattern corresponds to its rank. Thus, the null pattern at the top is always at rank 0. Lexical patterns are always at rank 1. Superlexical patterns are at ranks 2 to $k - 1$. This procedure defines the way a given instance such as (5) gets analyzed.

Now, let me explain the reverse operation of composition by showing how patterns are superimposed to produce instances. Consider superlexical patterns [—, faxed, Bill, a letter] (=p1), [Ann, —, Bill, a letter] (=p2), [Ann, faxed, —, a letter] (=p3), and [Ann, faxed, Bill, —] (=p4) for example. When they are unified along segments 1 to 4 in the way defined in Table 1, it produces instance (5) = p0. The procedure outlined here is detailed in Kuroda (2001) under the name of *Parallel Pattern Matching Analysis* (PMA).

Note, however, that this property of unifiability is recursive: p2, for example, is also a superposition of superlexical patterns q1, q2, and q3 in the way defined in Table 2. This implies that in a PL, any pattern or instance at rank k is defined as the unification of a set of patterns at rank $k - 1$.

2.3.2 Simulated Parallel Error Correction with Propagation Under the minimum introduction above, let us see how the semantic interpretation/meaning construction goes under a PL. I have the following basics of the model in mind (but unimplemented yet):

(11) Interpretation of e with Simulated Parallel Error Correction with Propagation (SPECPC):

Step 0: Assume that a PL $L(e)$ is constructed for e .

Step 1: List all the instances of the i th ($i \in k$) superlexical pattern at rank $k - 1$ and form sets I_1, I_2, \dots, I_k for each of them.

Step 2a: If the instance set for pattern p is empty, restart Step 2 with (superlexical) patterns at rank $k - 2$ of which p is an instance;

Step 2b: Otherwise, unify a tuple of as many instances of the k sets as possible.

Step 3: If unification is successful, equate the interpretation of e with that of the unified expression.

Let me explain this using a concrete case. Suppose that the interpretation of $p0 = [Ann, faxed, Bill, a letter]$ is attempted. In Step 0, a PL is given and we have $p1 = [—, faxed, Bill, a letter]$, $p2 = [Ann, —, Bill, a letter]$, $p3 = [Ann, faxed, —, a letter]$, and $p4 = [Ann, faxed, Bill, —]$. For simplicity, let us assume that there are no instances of utterance in which *faxed* is used as the verb of ditransitive construction. This implies: (i) The set of instances for $p1$ is an empty set. (ii) The set of instances for $p2$ is some set like { Ann **sent** Bill a letter, Ann **showed** Bill a letter, Ann **requested** Bill a letter, ... }. (iii) The set of instances for $p3$ is some set like { Ann faxed a letter, Ann faxed **a copy of** a letter, Ann faxed **and mailed** a letter, ... }¹⁵; and (iv) The set of instances for $p4$ is an empty set like $p1$. The meaning of [Ann, faxed, Bill, a letter] is equated with the one of [Ann, faxed and sent, Bill, a letter] if and only if [Ann, sent, Bill, a letter] obtained

¹⁴ With respect to this matter, I suspect that it is possible to interpret segmentation as an optimization over randomized samplings. If true, it could be simulated using the Monte Carlo method. An alternative is Bayesian learning (Mochihashi *et al.*, 2009).

¹⁵ The set has *Ann faxed a letter* in it as far as — can be null.

in Figure 3, three superlexical patterns, [Ann, —, Bill, a letter], [—, sent, Bill, a letter], and [Carol, sent, —, a letter], turn out to be relatively productive. At rank 2, two superlexical patterns, [—, Bill, a letter] and [—, sent, —, a letter], are relatively productive.

Some of the productive patterns are of the VP-type, but the others are not. They are not even constituents. This is not a shortcoming of PLM; rather, it is one of its advantages. PL is useful to recognize the relative productivity of such patterns, which a constituency-based analysis should dismiss.

2.4.1 No (need for) “constructional” meaning Crucially, productive patterns like [—, —, Bill, a letter] (or [Ann, —, Bill, a letter]) can dispense with *ditransitive construction* that Goldberg (1995) advocated to account for cases like (5) if only the variable of [Ann, —, Bill, a letter] (or [Ann, —, Bill, a letter]) has a strong selectional bias for verbs of sending. There is no need for the ditransitive construction to account for this bias, as long as we accept the VMT: such an effect is the natural property of the meaning construction under a huge lattice of instances illustrated in §2.3. It should be possible to show, through a corpus-based study or an experimental study, that even a less specified superlexical pattern [—, —, Bill, a letter] (or [Ann, —, Bill, a letter]) exhibits such selectional preferences. Under this, the stipulation of ditransitive construction in Goldbergian style is overkill, since PLM guarantees that the constructional meaning of a superlexical pattern P is just an “average” of all the meanings of the instances licensed by P . For example, the sense of caused possession usually attributed to ditransitive construction can be regarded as a by-product of superlexical units like [—, —, Bill, a letter] (or [Ann, —, Bill, a letter]). We need to ascertain the types of constructions that are possible and those are impossible, by formulating an explicit model of how linguistic memory is organized, but PLM is a promising candidate for it.

2.4.2 Importance of superlexical patterns Recall that a pattern at rank k is the defined unification of patterns at rank $k - 1$. Under this, it must be true that superlexical patterns closer to instances always contribute more to the interpretations of instances than lexical patterns at higher ranks do. This is the most straightforward account of why a garden variety of non-compositional units play a crucial role in the interpretation of the expressions of a natural language: to name a few, it includes “collocations” and “multi-word expressions” of which Sinclair’s idiom principle (Sinclair, 1991) holds, “constructions” that Construction Grammar (Croft, 2000; Goldberg, 1995) was devised for, and “formulas” in the sense of Wray (2002). I argue that PL is the simplest way to describe such phenomena.

If some exaggeration is allowed, it is possible to state that no meanings reside in words or phrases, but all the resources needed for interpretation reside in fully lexicalized instances of utterance. This is because under PL, patterns, lexical or superlexical, are just indices for instances, and they do not (need to) have meanings of their own. Thus, it can be argued that they appear to have meanings of their own just because they serve as “keys” to full instances that bear meanings. I do not claim this to be a fact, but I do not find it implausible. Crucially, there is a trade-off between precision and recall with patterns. The more instance variations or “types” a key has, the more ambiguous it gets. This justifies the fact that lexical patterns are worse predictors of sentential meanings than superlexical ones.

2.5 Need for refinement

2.5.1 Need for unsupervised segmentation The current implementation assumes that segmentation is given. This is clearly an ungrounded and opportunistic assumption. To validate the point of “no fixed lexicon” made in §2.2.5, it is necessary to implement a procedure for learning how to segment in an unsupervised fashion. The most promising way to do this would be the incorporation of unsupervised segmentation in the Bayesian framework (Mochihashi *et al.*, 2009).

2.5.2 Realistic treatment of variables As noted above, variables in a pattern are better understood as the memory-traces of lexical items or as “constants.” This implies that they encode both semantic and phonological constraints. This property is not taken into account in the version of PL described in this paper, and PLB has not implemented it as yet. This results in an obvious drawback. The current version of PLM is too admmissive in that it specifies the syntactic commonalities between two instances. Clearly, unconstrained abstractness for variables in patterns creates a source for this overgeneration. This, in fact, results in its failure to capture the constituency effects. If variable introduction is constrained, however, we can expect them to be properly described. This, of course, is left for future work.

2.5.3 Filtering out useless patterns The last point is related to the distinction between “useful” and “useless” patterns. The current implementation allows much room for useless patterns and lacks constraints to discard them. It is reasonable to expect that a fewer number of useless patterns will be recognized if the introduction of variables is constrained semantically. This is one of the various awaited improvements left for future work, since it requires a robust feature-handling system not implemented as yet.

3 Concluding remarks

There is a trade-off between the memory-based and grammar-based models, and therefore, a PL-based theory of syntax will always have certain shortcomings. The most obvious one is that if unconstrained, it allows for many “useless” patterns. This is evident when provided with arrays with many segments. It is observed that if an array has more than seven segments, the ratio of [useful patterns/useless patterns] drops drastically. In other words, the current implementation of the PL theory is only useful to capture the “local” dependencies. This implies that we will need an extra device to derive the so-called long-distance dependencies, which are hardly rare and exceptional in natural language. Considered from a different perspective, this motivates the assumption that human linguistic memory comes with a mechanism to avoid remembering useless patterns. It is still far from clear whether this is simply a matter of frequency or co-occurrence. Along with the properties of bounded productivity presented in §1.2, a natural language shows sensitivity to lower-frequency items and collocations that simple distributional statistics fails to capture. This aspect could be termed as the “mysterious survival of low-frequency items.” The grammar-based model has no shortcoming because it does not (need to) take frequency into account. A plain usage-based model, in which high-frequency of items is (implicitly) assumed to be a necessary condition on the formation of schemas or templates, is troublesome. In fact, if the formation of schematic knowledge was not to be constrained by frequency, what else could constrain it? However, this is not true of the PLM proposed in this paper. This suggests that PLM serves as a better model for the realistically usage-based model.

References

- Arbib, M. A., E. J. Conklin and J. C. Hill. 1987. *From Schema Theory to Language*. Oxford University Press.
- Barwise, J. and J. Perry. 1983. *Situations and Attitudes*. MIT Press.
- Bybee, J. L. 2001. *Phonology and Language Use*. Cambridge University Press.
- Chomsky, N. 1965. *Aspects of the Theory of Syntax*. MIT Press.
- Croft, W. 2000. *Radical Construction Grammar*. Oxford University Press.
- Daelemans, W. and A. van den Bosch. 2005. *Memory-based Natural Language Processing*. Cambridge University Press.
- Ganter, B., G. Stumme and R. Wille, editors. 2005. *Formal Concept Analysis: Foundations and Applications*. Springer.
- Goldberg, A. D. 1995. *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press.
- Hawkins, J. and S. Blakeslee. 2004. *On Intelligence: How a New Understanding of the Brain Will Lead to the Creation of Truly Intelligent Machines*. Times Books; Adapted edition.
- Hintzman, D. L. 1986. “Schema abstraction” in a multiple-trace memory model. *Psychological Review*, 93(4):411–428.
- Hopper, P. 1987. Emergent grammar. In *Berkeley Linguistics Society, Vol. 13*, pp. 139–157.
- Kuroda, K. 2001. Presenting the PATTERN MATCHING ANALYSIS, a framework proposed for the realistic description of natural language syntax. *Journal of English Linguistic Society*, 17:71–80.
- Langacker, R. W. 1988. A usage-based model. In Rudzka-Östyn, B., editor, *Topics in Cognitive Linguistics*, pp. 127–161. John Benjamins.
- Luria, A. R. 1987. *The Mind of a Mnemonist: A Little Book about Vast Memory*. Harvard University Press.
- McGaugh, J. L. 2003. *Memory and Emotion: The Making of Lasting Memories*. Columbia University Press.
- Milner, B., S. Corkin and H. L. Teuber. 1968. Further analysis of the hippocampal amnesic syndrome: 14-year follow up study of H.M. *Neuropsychologia*, 6.
- Mochihashi, D., T. Yamada and N. Ueda. 2009. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th Inter. Joint Conf. on NLP of the AFNLP*, pp. 100–108, Suntec, Singapore.
- Port, R. 2007. How are words stored in memory? Beyond phones and phonemes. *New Ideas in Psychology*, 25(2):143–170.
- Sinclair, J. M. 1991. *Corpus, Concordance, Collocation*. Oxford University Press.
- Tomasello, M. 2003. *Constructing a Language: A Usage-based Theory of Language Acquisition*. Harvard University Press.
- Wray, A. 2002. *Formulaic Language and the Lexicon*. Cambridge University Press.