

Introducing *Pattern Lattice Model* as a Form of *Extremely Usage-based Model*

Kow Kuroda

National Institute of Information and Communication Technologies (NICT), Japan

1 Introduction

This paper presents a brief summary of a model of human linguistic knowledge and performance called *Pattern Lattice Model* (henceforth PLM) proposed and developed by Kuroda and his colleagues (Kuroda 2009; 黒田 2009; 黒田 and 長谷部 2009; 吉川 2010a; 吉川 2010b). The purpose of this short paper is two-fold: the first aim is to clarify several points that makes PLM distinct. The second aim is to show that the basic tenets of Construction Grammar (henceforth, CG) (Croft 2000; Fillmore 1988; Goldberg 1995; Goldberg 2006) (e.g., priority of “constructional” meanings over “lexical” meanings) are natural consequences of an operation called *Unification over Parallel Simulated Error Correction* (UPSEC) assumed under PLM and therefore CG can be safely replaced by PLM.

2 What is the Pattern Lattice Model?

PLM is a form of **usage-based model** (UBM) of language (Langacker 1988), but it is more than just yet another form of UBM. It is a radically new form of UBM and even provides a radical reformulation of it. In more adequate terms, PLM is a **strongly example-based processing model** of language, though it also has something to do with grammar equated with “knowledge” of language distinguished from the processing mechanism of it. It is because PLM narrows possible forms of grammar drastically. I will return to this issue in §3.2.

2.1 Strongly example-based processing

PLM implements the idea of **strongly example-based language processing** (SEBLP). Example-based processing is a special form of **memory-based processing** (MBP). A processing is example-based if processing of new input are carried out using examples stored in memory. How is an example-based language processing carried out? The answer is straightforward. It is carried out roughly in the following way:

- (1) Suppose $E = \{e_1, e_2, \dots, e_N\}$ represents the set of all examples stored in memory. Given an input t ,
 - a. find a set of examples $E' = \{e_1, e_2, \dots, e_n\}$, a subset of E , that consists of all and only examples similar to t with distance measure d . We say e_i is “close enough” to t if $0 \leq d(t, e_i) \leq d_\theta$ where d_θ is a parameter that specifies the threshold value. If the distance is standardized to “similarity” measure $s(t, e_i)$, we have $0 \leq s(t, e_i) \leq 1$.
 - b. select $E'' = \{e'_1, e'_2, \dots, e'_k\} (\in E')$ such that the semantics of e'_i and e'_j are compatible.
 - c. equate the semantics of t as the logical disjunction (or average) over the semantics of E'' .
- The degree of dependence on stored examples determines the strength of memory-basedness. A processing is strongly memory-based if processing of input depends on stored examples more than abstract(ed) structures like schemas.

2.2 The full memory hypothesis

What makes PLM distinct from many other theories of language is the following hypothesis:

- (2) **The full memory hypothesis:** A speaker of a language L commands a “full memory” of linguistic experience (in implicit memory distinguished from “explicit memory (Milner, Corkin, and Teuber 1968)) to process any expressions of L .

Admittedly, this is a controversial hypothesis. In fact, no theory of language accepted in linguistics seems to take this very seriously, but PLM dares to accept it, at least strategically,¹⁾ The reason is that, as I show, its acceptance offers theoretical linguistics more gains than losses.

2.3 How to deal with interpretation

One of the most serious challenges to strongly memory-based models like PLM is the seeming compositionality of natural language semantics. It is often claimed that it requires effective syntax and that memory-based models are too weak to provide it. Its intent is that such effective syntax is provided only by grammar.

The plausibility of this claim, however, can be illusory. Such a claim can be invalidated only when memory-based models are shown to be able to deal with seeming compositionality of natural language semantics. Unification over Parallel Simulated Error Correction (UPSEC) was designed for this purpose. It determines the interpretation of input t roughly in the following way:²⁾

- (3) a. If there is t' stored in memory such that $t = t'$, equate t 's semantics with t' 's semantics.

¹⁾The reason that PLM accepts the full memory hypothesis is that it was developed as a theoretical extension of Robert Port's *Rich Phonology* presented in Port (2007, 2010). See Kuroda (2009) for relevant discussion.

²⁾This paper only deals with its essential features. See Kuroda (2009) for more details of UPSEC.

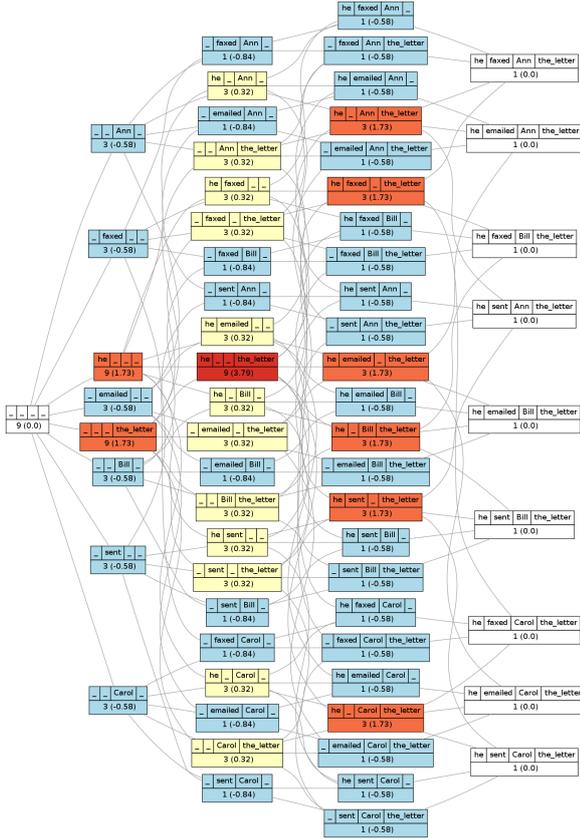


Figure 1: Pattern lattice over the examples in (5).

- b. If not, equate the semantics of t with the average³⁾ of the semantics of $E^t = \{e_1, e_2, \dots, e_n\}$ such that $e_i \in E$ is close enough to t .

under the following principle:

- (4) Principle of distributed resource allocation: for every x_i of a given instance $e = [x_1, x_2, \dots, x_n]$ consisting of n segments, identify resources r_1, r_2, \dots, r_n for interpretation using patterns as close to e as possible and unify r_1, r_2, \dots, r_n .

The crucial point is how to calculate e_i 's closeness to target t , or **how to define a similarity measure between a given pair of forms**. A pattern lattice over instances defines a good metrics for this similarity measure.⁴⁾

Figure 1 diagrams the lattice over the 9 examples in (5) which consists of examples matching pattern $SV O_1 O_2$.⁵⁾

- (5) (a) he faxed Ann the.letter; (b) he emailed Ann the.letter; (c) he sent Ann the.letter; (d) he faxed Bill the.letter; (e) he emailed Bill the.letter; (f) he sent Bill the.letter; (h) he faxed Carol the.letter; (i) he emailed Carol the.letter; (j) he sent Carol the.letter

³⁾Traditional linguistics does tell us what the “average” of semantics is, but it can be easily specified as a logical disjunction if meanings are represented as feature vectors.

⁴⁾There are additional minor assumptions, however. It is assumed that each edit has the same amount of cost, which need not be generally true.

⁵⁾The diagram was generated by pattern lattice builder developed by Yoichiro Hasebe (Doshisha University) and made freely available at <http://www.kotonoba.net/rubyfca/pattern>.

Every pattern in pattern lattice is assigned a “rank.” Roughly, the rank of a pattern is equal to the number of lexically realized segments. Thus, the leftmost pattern, $[_, _, _, _]$, is at rank 0. Patterns on the second column from left, $[he, _, _, _]$, $[_, faxed, _, _]$, $[_, emailed, _, _]$, $[_, sent, _, _]$, $[_, _, Ann, _]$, $[_, _, Bill, _]$, $[_, _, Carol, _]$, and $[_, _, _, the\ letter]$, are at rank 1, and so on.

We always have only 1 pattern at rank 0, $[_, _, _, _]$. This is called the “top” (\top) of the pattern lattice. The pattern at rank 0 encodes the co-occurrence information of a given set of instances in the most abstract way. Patterns with no lexical specification are called either “nonlexical pattern” or “abstract patterns.”

We have 9 patterns at rank 1, i.e., $[he, _, _, _]$, $[_, faxed, _, _]$, $[_, emailed, _, _]$, $[_, sent, _, _]$, $[_, _, Ann, _]$, $[_, _, Bill, _]$, $[_, _, Carol, _]$ and $[_, _, _, the\ letter]$, are called “lexical patterns” in contrast with “superlexical patterns” defined below.

We have 22 patterns at rank 2 and 24 patterns at rank 3, respectively. At rank 3, we have: $[he, faxed, _, _]$, $[he, emailed, _, _]$, $[he, sent, _, _]$, ... At rank 4, we have: $[he, faxed, Ann, _]$, $[he, emailed, Ann, _]$, $[he, sent, Ann, _]$, ... These patterns are called “superlexical/supralexical patterns” in contrast with “lexical patterns” defined above.

In a pattern lattice, a pattern at rank k is an abstraction over a set of either instances or patterns at rank $k + 1$. This holds for every k recursively. In general, patterns at rank k have k lexical items. Patterns at rank k for instances with n segments are called (i) nonlexical when $k = 0$; (ii) lexical when $k = 1$; and (iii) superlexical when $1 < k < n$.

Note that a pattern lattice defined in this way specifies, in a natural and unambiguously way, the **accessibility hierarchy for resources needed for interpretation**. A nice thing about this is that it enables us to unify noncompositional and compositional modes of interpretation. Purely compositional interpretation consists of the lexical semantics specified by lexical patterns at rank 1. They are more or less compositional because these patterns have just one lexical item and do not specify higher-order, co-variational semantics among multiple items. If the exploitation of patterns at rank k are, by definition, always less preferred by that of patterns at rank $k + 1$, it follows that the semantics of lexical patterns are always the “last resort.” Clearly, this is a definition that unifies the compositional and noncompositional modes of semantic interpretation.

2.4 How UPSEC works under PLM

It should be clear now that a pattern lattice is useful to determine the set of similar instances. Suppose, for example, that (5b) = *he emailed Ann the letter* is a new input recognized for the first time. This means that no semantic information in the memory is available for its meaning. To construct it, the system does the following by accessing a set of examples via four superlexical patterns at rank 3, i.e.,

- (6) a. $p_1 = [_, emailed, Ann, the\ letter]$,
 b. $p_2 = [he, _, Ann, the\ letter]$,
 c. $p_3 = [he, emailed, _, the\ letter]$, and
 d. $p_4 = [he, emailed, Ann, _]$

which are generated via SPEC. Assume additionally that p1, p3 and p4 have no other instances than (5b).⁶⁾ In this case, the only source available for approximation of the semantics of (5b) is to average the semantics of the instances of p2, i.e., [he, faxed, Ann, the letter] and [he, sent, Ann, the letter] which are at the leaves⁷⁾ of the lattice in Figure 1.

2.5 No need for constructions *per se*

In a narrow sense, PLM is a model of linguistic forms. PLM is not a self-contained theory of language in this specific sense. But it is quite straightforward to see how semantics works in PLM as far as we assume that every form is stored in couple with its meaning.⁸⁾

Note that the PLM-based description of the interpretation of (5b) can replace what Goldberg (1995, 2006) purported to account for in terms of her “ditransitive construction.” It is interesting to see that she tries to attribute constructional meaning to [_, _, _, _] at rank 0, corresponding to *SVO₁O₂* or *NPVNP*, which is **more abstract than lexical patterns at rank 1**. This is exactly why Goldberian account cannot be free from overgeneralizations.

By contrast, UPSEC describes the same set of phenomena without positing any constructions, and I claim that this is exactly what makes PLM more explanatory and CG. I want to discuss this issue in more detail.

CG is a linguistic theory that attracts many researchers. One of its crucial assumptions is the independence of “constructional” meanings from “lexical” meanings. But, as we saw in the presentation above, PLM accounts for the same phenomenon without positing constructions and the like. All what PLM assumes for this is the following:

- (7) a. Semantic interpretation of input *t* is determined either by a direct memory access to *t*'s meaning stored in memory or by a “transfer” from the semantics of examples similar enough to *t*.
- b. A pattern lattice specifies the range of accessibility required to determine what examples count as “similar enough.”

So, we may ask, What relation can PLM have to CG? Roughly speaking, there are three possibilities: (i) PLM is one of the garden variety of CG (PLM as a variation of CG), (ii) PLM and CG are different frameworks but they supplement each other's weaknesses (PLM as a supplement to CG), and (iii) CG is a derivative of, if not a variant of, PLM that all stipulations and predictions from CG are derivable from PLM (PLM as a replacement to CG). I believe the argument so far provided strong evidence for (iii).

In the view of PLM, constructions are best characterized as derivatives of a huge pattern lattice that is automatically and blindly constructed over a set of instances.⁹⁾ Note that a pattern lattice specifies all possibilities for schematic representations at all possible levels of abstractness. In short, **PLM knows no distinction between constructions and**

⁶⁾Note that this assumption is natural because it exactly corresponds to the fact that *email* was never used in the ditransitive construction.

⁷⁾Usually, the bottom (⊥) of a pattern lattice is not drawn to save space.

⁸⁾In fact, it is unnatural to assume that a speaker stores a form and a meaning independently, without any interconnectedness, when he or she stores a piece of speech *s*.

⁹⁾This paper does not deal with the question what constructs a pattern lattice. This is an open question.

nonconstructions as far as forms are equated with their semantics. But does this mean that PLM is insufficient? I argue not. First of all, what are constructions after all? First of all, there is no known clear-cut boundary between constructions and nonconstructions. In a sense, **identification of constructions is a matter of degree**. If so, the real problem is how to measure the degrees of goodness for the identification of constructions. Note that this is exactly the same kind of problem that we have in the PLM. In other words, CG can never be better than PLM unless it has an explicit mechanism for automatic identification of constructions, which I claim is not satisfied with any version of CG.

If we assume PLM, some of the crucial tenets such as the following automatically follow:

- (8) a. The crucial assumptions (e.g., independence of “constructional” meanings to “lexical” meanings) and theoretical predictions (e.g., acceptability pattern accounted for in terms of particular constructions) that make CG explanatory and attractive to the linguistics community are plain consequences of PLM. This means that CG need not be assumed as far as we accept PLM.
- b. PLM makes nontrivial predications (e.g., that construction effects are distributed and you can not usually single out a construction that is solely responsible for a particular phenomenon, that different constructions have different degrees of “constructionhood”) that are not directly available in CG.
- c. PLM can model several aspects of human language (e.g., formulaicness) that CG cannot.

In the view of PLM, constructions are part of side-effects of the USPEC under a pattern lattice. If the goal of linguistics is to identify “constructional meanings” as distinct from lexical meanings and describe them properly, UPSEC under PLM should suffice. In other words, **linguists do not need to say explicitly that constructions (should) exist when they want to account for construction effects**.

PLM does not say, however, that there are no constructions whatever. They should exist, but **what is really challenged by PLM account is how to identify constructions**. The point is that linguistics will never make any remarkable achievement if constructions are identified using linguists' intuition alone.

3 Discussion

3.1 Limitations of PLM

No matter how many benefits are associated with it, PLM cannot be without limitations, if not shortcomings, of its own, at least in the current implementation. Let me specify a few of them.

First of all, PLM makes sense only when we accept the full memory hypothesis. This position is rather costly because it is not so easy to defend such position, though Kuroda (2010) offers several arguments for this. So, it would be safer to admit that PLM can happen to rely on a possibly unfalsifiable assumption. In fact, it is an open question if **all** utterances perceived, rather than **most** of them, are stored in (implicit) memory. The crucial point,

however, is that even if they are not all what people heard and understood, **human linguistic performance is very likely to rely on a tremendous amount of exemplar memory than most linguists are willing to admit**, and that this requires human linguistic competence to take another form than generative linguists tend to (hastily) assume. This is what the full memory hypothesis really means.¹⁰⁾

Second, the full memory based modeling is, as expected, computationally quite demanding. This is more serious about the number of segmentation. For example, the computational complexity increases exponentially with the number of segments. For example, when a pattern has more than 7 segments, the overall computational time gets nearly unacceptable. Compared to this, the impact of the number of instances is much milder. The increase of computational cost is just logarithmic. We must admit that it is an open question why search for examples can be so fast in human language processing.

Third, as stated above, no reliable statistical measure is developed to differentiate “good” patterns from “bad” patterns. In the current implementation, *z*-score is tentatively used for this purpose, but there are several problems associated with it. One of them is that rank-relative productivities of patterns are not guaranteed to obey the Gaussian distribution. If this assumption is not met, it is inadequate to make use of *z*-score to represent rank-relative productivity: *z*-score is used only because no better alternative is known. We will need to develop a sophisticated, probably more complex measure to achieve more natural results.

Another issue associated with this is that it is not fully accounted for what impact speech errors have on language processing. The full memory hypothesis forces us to assume that incorrect forms of utterance are stored in implicit memory as well as correct ones, but this sounds somewhat ridiculous. We will need to introduce a mechanism that guarantees incorrect forms of utterance cause as little harm as possible.

Finally, one of the most obvious limitations of PLM is that it provides no straightforward mechanism to deal with so-called syntactic alternations. Even simple alternations like question formation require certain tricks. To many linguists, this may look disappointing because they can be easily captured with syntactic movements, but we have a trade-off and it is not easy to tell, at least for the moment, if this is really a bad thing.

3.2 PLM-compatible forms of grammar

PLM frees linguists from the burden for identification of constructions. It is suggested that constructions are epiphenomenal. By the same token, PLM places a far lesser explanatory importance to “schemas.” The reason is roughly that, under the full memory hypothesis, schemas are merely indices of instances stored in the memory. They do not “sanction” or “license” any new instances. What they do is two-fold. For positive instances, they determine the sets of instances available for the calculation of the semantics of given inputs. For negative instances, schemas block their access to instances potentially available for the semantic computation.

What does this mean after all? To put it most crudely, schemas are not an explanatory concept any more, at least unless we already tell exactly what examples are instances of what schemas. I’m afraid this condition is not satisfied at all in the current situation of Cognitive Linguistics/Construction Grammar. I believe this is why it fails to replace Generative Linguistics.

References

- Croft, W. (2000). *Radical Construction Grammar*. Oxford: Oxford University Press.
- Fillmore, C. J. (1988). The mechanisms of ‘Construction Grammar’. In *BLS*, Volume 14, pp. 35–55. BLS.
- Goldberg, A. D. (1995). *Constructions: A Construction Grammar Approach to Argument Structure*. Chicago, IL: University of Chicago Press.
- Goldberg, A. E. (2006). *Constructions at Work: The Nature of Generalizations in Language*. Oxford: Oxford University.
- 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. [邦訳『考える脳 考えるコンピューター』(伊藤文英訳). ランダムハウス講談社.]
- Kuroda, K. (2009). Pattern lattice as a model for linguistic knowledge and performance. In *Proceedings of the 23rd PACLIC, Vol. 1*, pp. 278–287. <http://clsl.hi.h.kyoto-u.ac.jp/~kkuroda/papers/kuroda-paclic23-paper.pdf>.
- Langacker, R. W. (1988). A usage-based model. In B. Rudzka-Östyn (Ed.), *Topics in Cognitive Linguistics*, pp. 127–161. Amsterdam/Philadelphia: John Benjamins.
- 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.
- Port, R. (2007). How are words stored in memory? beyond phones and phonemes. *New Ideas in Psychology* 25(2), 143–170.
- Port, R. (2010). Rich memory and distributed phonology. *Language Sciences* 32(1), 43–55.
- 黒田 航 (2009). パターンのラティス下での疑似並列エラー修復に基づく文意の構築. In 日本認知科学会第 26 回大会発表論文集, pp. 236–237.
- 黒田 航 (2010). 超常記憶症候群の理論的含意. In 日本認知科学会第 27 回大会発表論文集.
- 黒田 航 and 長谷部 陽一郎 (2009). Pattern Lattice を使った (ヒトの) 言語知識と処理のモデル化. In 言語処理学会第 15 回大会発表論文集, pp. 670–673.
- 吉川 正人 (2010a). 「語」を越えた単位に基づくコーパス分析に向けて: パターンラティスモデル (plm) とその有用性. *藝文研究* 98, 221–207.
- 吉川 正人 (2010b). パターンの生産性に見る統語発達: パターン束モデルに基づく習得プロセスの検証. In 認知科学会第 27 回大会発表論文集.

¹⁰⁾Incidentally, a view on artificial intelligence and cognitive science fully compatible with this thesis is presented and compellingly argued for in works by Jeff Hawkins such as (2004).