THE MAGIC NUMBER 4:
EVOLUTIONARY PRESSURES ON SEMANTIC FRAME STRUCTURE

DEKAI WU

HKUST, Human Language Technology Center, CSE Department, Hong Kong

dekai@cs.ust.hk

We propose that the "magic number 4" puzzle of argument structure in semantic frames across most human languages is explained by selection pressures given inherent computational efficiency properties naturally arising from fundamental combinatorial mathematics of compositionality. Irrespective of language, school, or theoretical bias, linguists have long observed that what are now generally called "semantic frames" empirically bear a maximum limit of four core arguments per frame, for unknown reasons. We explain how this limit would automatically emerge as a consequence of evolutionary preference for the formal equivalence class of *inversion transductions* as an optimal balance between expressivity and fast tractable polynomial-time language learning and interpretation/transduction between different representation languages.

1. Introduction

The puzzle of the magic number 4 has escaped explanation since the 1960s, when various lexically oriented theories of what have come to be generally called "semantic frames" emerged. Since then, dozens if not hundreds of lexical semantics and "deep syntax" models have been formalized, with a high degree of terminological variance that frequently obscures their mathematical commonalities. Although details differ between theories, all draw the distinction between core arguments of the predicate (complements) and optional arguments that provide various classes of auxiliary information but can be omitted without affecting the remainder of the sentence (adjuncts).

The odd thing is that these all these semantic frame theories permit no more than four core argument roles per frame. Depending on the theory, these core arguments—or "semantic roles"—may be labeled "agent", "patient", "experiencer", "recipient", and so on. Historically, many theories cite a motivation for semantic roles as "deep cases" as opposed to the surface syntactic cases that are marked in many languages (nominative, accusative, dative, genitive, etc.) Across all such theories, the fact that semantic frames do not admit more than about four core arguments has remained remarkably consistent over the decades.

In his seminal work on Functional Grammar, Dik (1978) observed that "It is an empirical fact that nuclear predications in natural languages hardly ever have more than four different argument positions. Even predicates with four nuclear arguments are rare in any language. This puts interesting constraints on the variety of predicate-frames to be found in the lexicon of any language."

This observation is borne out by the comprehensive survey of case and valency of Somers (1987), who reviewed numerous systems of semantic frames, across a broad range of languages and schools of linguistic theory, from the landmark Case Grammar of Fillmore (1968) and verb valency in dependency grammars (Tesnière, 1969), to many other related predicate-argument structures such as the theta roles or thematic relations of GB (Chomsky, 1981). *None* of the lexical semantics systems reviewed by Somers ever suggested a frame with more than four participants.

The same empirical phenomenon is found in the most recent incarnations of lexical semantics theory, from FrameNet (Baker *et al.*, 1998) to construction grammar in the various forms of Lakoff (1987), Fillmore *et al.* (1988), Croft (2001), Goldberg (2006), or Steels (2012). The widely used PropBank semantic frame annotated corpus (Palmer *et al.*, 2005)—which, in an attempt to remain as theory-agnostic as possible, eschews historic semantic role names like "agent" or "actor" or "nominative" in favor of the generic ARG0, ARG1, ARG2, and so on—rarely sees a frame with an ARG3, and almost never an ARG4 or ARG5.

Yet despite this surprisingly consistent phenomenon across decades of semantic theory—from different schools of linguistics around the world, following different empirical methodologies and theoretical principles, studying many different languages—few, if any, have asked where this seemingly universal magic number 4 comes from, or why it would have arisen cross-linguistically. Why should languages around the world, stemming from vastly different origins, all have evolved to converge on the same limit of four core arguments per semantic frame?

2. Evolutionary pressures on semantic frame structure

We propose here that the mathematics of **inversion transduction grammars** or ITGs (Wu, 1997) naturally explain how language evolution would drive semantic frame structure to converge upon the magic number 4. Our logic rests on fundamental combinatorial principles and computational complexity properties of algorithms over different equivalence classes of transductions. In classic formal language theory, a **transduction** is a relation between two languages, or in other words, a set of sentence pairs. This is the bilingual generalization of a **language**, which is a set of sentences. A **transduction grammar** or **translation grammar** (TG) is a bilingual grammar that generates a transduction, by describing the relations between two languages in terms of how smaller units compose into larger units. Naturally, the components of different languages may have to be ordered differently, which means that transduction grammars must have some way to describe the *permutation* of components.

A **transduction rule** is a formalization of a construction, or Saussurean sign. Whatever the type of transduction grammar, one can always try to describe more complex constructions via assemblies of multiple transduction rules. The archetypical type of TGs are **syntax-directed transduction grammars** or SDTGs, introduced by Lewis and Stearns (1968) and Aho and Ullman (1972).[a] SDTGs generate **syntax-directed transductions** or SDTs, for which there are no polynomial time algorithms for solving the recognition (bilingual parsing) problem.

Formally, an SDTG is a tuple $\langle N, \Sigma, \Delta, R, S \rangle$, where N is a finite nonempty set of nonterminal symbols, Σ is a finite set of terminal symbols in the input language L_0, Δ is a finite set of terminal symbols in the output language L_1, R is a finite nonempty set of transduction rules and $S \in N$ is a designated start symbol. There are many variant notations, but a simple and unambiguous formalization restricts a **syntax-directed transduction rule** to take the form:

$$A \to \Psi^{+}; \pi_0, \pi_1, ..., \pi_{k-1}$$

where $A \in N$ is a nonterminal symbol, Ψ^{+} is a nonempty sequence of k nonterminals and biterminals, and $\pi_i \in \{0, ..., k-1\}$ is an element in a **permutation vector** that specifies where Ψ_i is to placed after transducing. A **biterminal** is a pair of symbol strings: $\Sigma^* \times \Delta^*$, where at least one of the strings have to be nonempty.

Some twenty years ago we turned our statistical NLP research toward attacking multilingual machine learning and statistical machine translation (SMT) problems because, like many linguists, we believe theories of language should be rooted in cross-linguistic explanatory power. In the course of our work, we introduced the strongly restricted subclass of SDTs known as **inversion transductions**, and developed stochastic versions of phrasal transduction grammars along with efficient algorithms for (a) learning stochastic phrasal ITGs from parallel training corpora, (b) parallel data analysis, segmentation, parsing, and cross-linguistic projection, as well as (c) direct translation or decoding (Wu, 1997, 2010, 2014; Saers *et al.*, 2012; Saers and Wu, 2013). The empirical effectiveness of their inductive biases have placed inversion transductions at the foundations of most current state-of-the-art SMT systems including, e.g., Moses (Koehn *et al.*, 2007) and Hiero (Chiang, 2007) which induce forms of ITGs and translate using ITG decoders.

In modeling cognition, it is worth noting that all perception and interpretation is a form of translation. Merely being able to parse the syntactic structure of an input is pointless from the standpoint of evolutionary advantage; what makes perception and interpretation useful is the *translation* of the input into an alternative representation or language that allows the input to be dealt with as effortlessly as possible—for example, transducing an input utterance into a semantic frame representation. (Note that both the input utterance and the semantic frames can always be represented as strings; a nested compositional structure can trivially be (de)serialized from or into linear form via ordering and explicit markers.)

[a] A later synonym for transduction grammars used only in computational linguistics, "synchronous grammars", is not as widely recognized throughout mathematics and computer science. Similarly, "synchronous CFG" is synonomous with "SDTG". As explained in Section 3, however, the closest bilingual generalization of CFGs is ITGs, rather than SDTGs or synchronous CFGs.

This view predicts that selection pressures tend to drive both surface structure and semantic frames to evolve toward those classes that are not merely sufficiently expressive but also efficiently *transduceable*. A language that can be interpreted (transduced) quickly into a useful semantic representation will grant its users competitive advantages both (a) against communities using less easily transduceable languages and semantic frames, and (b) in the face of environmental adversity and competing species. Similarly, selection pressures will prefer transductions (surface structures and semantic frames) that are efficiently *learnable*.

How can "efficiently" be meaningfully defined, however? Herein lies the advantage of viewing semantic frames in terms of transduction from input sentences. Just as with monolingual languages, formal language theory categorizes transductions into different equivalence classes of generative capacity (expressiveness), normal forms, and computational complexity for various standard types of (bilingual) parsing, translation, and induction algorithms. This lets us empirically compare tradeoffs between expressiveness and efficiency for different equivalence classes of transductions, at an abstract computational complexity level that is agnostic as to specific symbolic or neural hardware realizations.

Our other motivation for attacking the SMT task was that very large quantities of relatively reliable parallel training data could be obtained, even for Chinese and English which are extremely different languages. Although this is not quite the same thing as having a parallel corpus of Chinese sentences and their semantic frame representations, requiring accurate transduction from Chinese directly to English is, if anything, even more challenging than to an intermediate semantic representation and then from there to English. In other words, we approximate the semantic frame representation using an English description of it. If we are able to come close to solving the harder direct transduction problem, then any empirical results on representational adequacy should be transferable to models that transduce to semantic frame representations as well.

3. Complexity for different classes of transductions

Some forms of translation are easier than others. For SDTGs, algorithms for the recognition problem are exponential time, which is clearly not an evolutionary advantage. However, by strongly restricting ourselves to narrower subclasses of syntax-directed transductions, it turns out that polynomial time algorithms become possible. We discuss only the most salient aspects of the formal analyses; for full details of various aspects the reader is referred to Wu (1997, 2010, 2014).

A restriction to finite-state transductions is clearly too strong. Algorithms for recognition of FSTs are very fast (linear time). However, FSTGs generate both the input and output strings in the same monotonic left-to-right order, and thus cannot express any reordering permutations at all.

However, inversion transductions empirically turn out to be an expressive yet efficient way to model translation. Unlike FSTGs, ITGs allow the components of

Table 1. Growth in number of alignment permutations for a sequence of length n.

n	**linear transductions**	**inversion transductions**	**syntax-directed transductions (*n!*)**	**ratio**
0		1	1	1.000
1	1	1	1	1.000
2	2	2	2	1.000
3	6	6	6	1.000
4	20	22	24	0.917
5	68	90	120	0.750
6	232	394	720	0.547
7	792	1,806	5,040	0.358
8	2,704	8,558	40,320	0.212
9	9,232	41,586	362,880	0.115
10	31,520	206,098	3,628,800	0.057
11	107,616	1,037,718	39,916,800	0.026
12	367,424	5,293,446	479,001,600	0.011
13	1,254,464	27,297,738	6,227,020,800	0.004
14	4,283,008	142,078,746	87,178,291,200	0.002
15	14,623,104	745,387,038	1,307,674,368,000	0.001
16	49,926,400	3,937,603,038	20,922,789,888,000	0.000

different languages to be ordered differently. But unlike SDTGs, rather than allowing arbitrary reordering and paying the price of exponential time complexity, ITGs impose restrictions that cut the computational complexity down to a manageable polynomial. A mathematically remarkable property is that three alternative restrictions all provably yield the same equivalence class of inversion transductions: (a) only transduction rules of rank 2 are permitted (no more than two nonterminals on the right-hand-side), or (b) only transduction rules of rank 3 are permitted (no more than three nonterminals on the RHS), or (c) only monotonically straight or inverted permutations are permitted (only left-to-right or right-to-left reordering).

The last alternative definition can be formalized by restricting inversion transduction rules to take one of the following forms:

$$S \to [A]\,,\ A \to \left[\Psi^{+}\right],\ A \to \langle\Psi^{+}\rangle$$

where the square and angled brackets denote straight and inverted order respectively. With straight order, both the L_0 and L_1 are generated left-to-right, whereas with inverted order, L_1 is generated right-to-left.

Unlike SDTGs, ITGs also have a 2-normal form, analogous to Chomsky normal form for CFGs, where the rules are restricted to only the following forms:

$$S \to A,\ A \to [BC]\,,\ A \to \langle BC\rangle,\ A \to e/f$$

where $A, B, C \in N$ are nonterminal symbols, and e/f is a biterminal string.

Between finite-state and inversion transductions, we have also recently introduced and empirically studied the class of **linear transductions**, as a bilingual analog of linear languages. A **linear transduction grammar** or LTG is restricted to only transduction rules of rank 1 (Saers *et al.*, 2011).

Table 1 compares how the number of permutations grows with the length of the sequence being transduced, for linear vs. inversion vs. syntax-directed transduc-

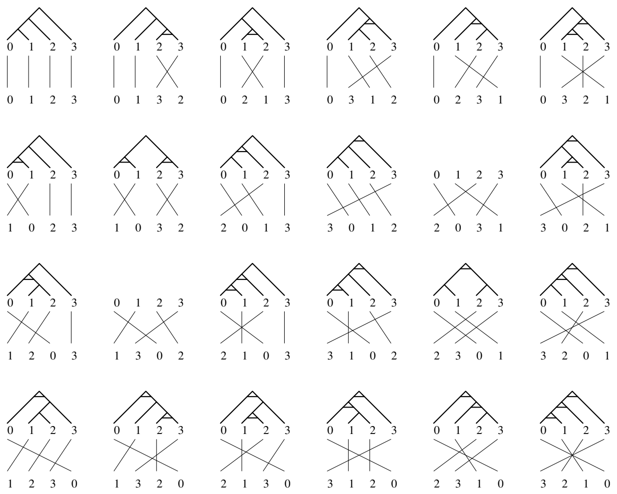

Figure 1. The 24 permutations of length 4, with 2-normal form ITG parses for 22. Nonterminal and terminal labels are omitted. A horizontal bar under a parse tree node indicates an inverted rule.

tions. For inversion transductions, the series is the Schröder numbers. Note that we have perfect coverage up to permuting three elements, nearly perfect coverage for four elements (22 out of 24 permutations), 75% coverage for five elements, 55% coverage for six elements, then dropping off rapidly for longer sequences.

Let us examine the inflection point, at four elements. Figure 1 depicts all 24 possible permutations of four elements (which can be constituents, phrases, or words), along with ITG parse trees to generate 22 of them. The only two permutations that cannot be generated are, in fact, the same case turned upside down.

Consider what this odd $(2, 0, 3, 1)$ permutation does. Given two adjacent elements 1 and 2, it not only inverts them, but then moves them even further from each other, to outside *both* sides of the surrounding context. Intuitively, this seems like a strange thing to do when trying to preserve meaning across a translation.

In our initial work on ITGs (Wu, 1997), we tested this upon the Hong Kong Hansard parallel Chinese-English corpus. Not only did the $(2, 0, 3, 1)$ permutation never occur, but *all* of the other 22 permutations did occur. This is a somewhat remarkable match between a mathematical model and a large empirical dataset.

Since then, a very large range of experiments have been conducted by many groups worldwide, across dozens of language pairs. Even though most "gold standard" test sets rely on heuristic automatic alignment programs and/or datasets containing flawed translations—and therefore actually underestimate the coverage of ITGs, as in for instance the study by Zens and Ney (2003) on German and French—nevertheless the coverage of inversion transductions remains almost universally extremely high. Wu (1997) gives concrete example sentence pairs illustrating why

MONOLINGUAL		BILINGUAL	
regular or finite-state languages **FSG (FSA)** *CFG that is right linear or left linear*	$O(n)$	regular or finite-state transductions **FSTG (FST)** *SDTG that is right linear or left linear*	$O(n)$
linear languages **LG** *CFG that is unary or linear*	$O(n^2)$	linear transductions **LTG** *SDTG that is unary or linear*	$O(n^4)$
context-free languages **CFG**	$O(n^3)$	inversion transductions **ITG** *SDTG that is binary or ternary or inverting*	$O(n^6)$
		syntax-directed transductions **SDTG** *(or synchronous CFG)*	$O(n^{2n+2})$

Figure 2. Hierarchy of equivalence classes and their complexities, for languages vs. transductions.

ITGs provide surprising flexibility in permutations.

Finite-state, linear, inversion, and syntax-directed transductions actually form a hierarchy of equivalence classes for transductions analogous to Chomsky's hierarchy for languages, as shown in Figure 2 together with their computational complexity for the standard dynamic-programming recognition algorithms that underlie parsing and EM learning. Unlike the monolingual case, no 2-normal form exists for SDTGs. Just as in the monolingual case, there is a tradeoff between generative capacity and computational complexity: the more expressive classes of transductions are orders of magnitude more expensive to biparse and train.

4. Discussion and conclusion

Unlike the monolingual case, ITGs represent the most expressive equivalence class still having tractable polynomial-time complexity. It is this property that allows us to run large-scale SMT experiments in unsupervised induction of transduction grammars. Empirically, even SMT systems that start out with unrestricted SDTG representations are nearly always simplified down to ITGs because doing so empirically results in significantly higher translation accuracy—indicating a better matched inductive bias at the level of inversion transductions.

In contrast to measurements on the permutations of surface form strings, Addanki *et al.* (2012) recently directly measured LTG vs. ITG coverage of cross-linguistic *semantic frame* alternations using annotated parallel PropBank data. Not only did ITGs cover 100% of the semantic frame alternations across languages, but even LTGs somewhat surprisingly covered 97%.

We have proposed that the magic number four is explained via evolutionary pressures of efficient transduceability and learnability, stemming from fundamental combinatorial properties of inversion transductions that simultaneously (a) generate frame-like Saussurean sign structures expressing almost any transposition of up to about four arguments, (b) match a wide range of cross-linguistic sequence reordering data, and (c) admit tractable polynomial-time recognition algorithms for both language interpretation and learning. A potential implication is that since semantic frame and construction grammar formalisms today exceed inversion transduction generative capacity, they may have too much expressive power to be efficiently learnable and an ITG restriction could provide a better learning bias.

Acknowledgments

This material is based upon work supported in part by DARPA under BOLT contract HR0011-12-C-0016, GALE contracts HR0011-06-C-0022 and HR0011-06-C-0023; by the EU under the FP7 grant agreement 287658; and by the Hong Kong RGC research grants GRF620811, GRF621008, and GRF612806. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC.

References

Karteek Addanki, Chi-Kiu Lo, Markus Saers, and Dekai Wu. LTG vs. ITG coverage of cross lingual verb frame alternations. In *EAMT-2012*, Trento, Italy, May 2012.

Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation, and Compiling (Volumes 1 and 2)*. Prentice-Hall, Englewood Cliffs, NJ, 1972.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *COLING-ACL'98*, Montreal, Aug 1998.

D. Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, 2007.

Noam Chomsky. *Lectures on Government and Binding*. Mouton, 1981.

W. Croft. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford, 2001.

Simon C. Dik. *Functional Grammar*. North-Holland, Amsterdam, 1978.

Charles J. Fillmore, Paul Kay, and Catherine O'Connor. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, 64:501–538, 1988.

Charles J. Fillmore. The case for case. In Bach and Harms, eds., *Universals in Linguistic Theory*, pp. 1–88. Holt, Rinehart, and Winston, 1968.

Adele Goldberg. *Constructions at Work: The Nature of Generalization in Language*. Oxford, 2006.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL-2007 Demo and Poster Sessions*, Prague, Jun 2007.

George Lakoff. *Women, Fire, and Dangerous Things*. University Of Chicago, 1987.

Philip M. Lewis and Richard E. Stearns. Syntax-directed transduction. *Journal of the Association for Computing Machinery*, 15(3):465–488, 1968.

Martha Palmer, Paul Kingsbury, and Daniel Gildea. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–105, Mar 2005.

Markus Saers and Dekai Wu. Bayesian induction of bracketing inversion transduction grammars. In *IJCNLP 2013*, Nagoya, Japan, Oct 2013.

Markus Saers, Dekai Wu, and Chris Quirk. On the expressivity of linear transductions. In *MT Summit 2011*, Xiamen, China, Sep 2011.

Markus Saers, Karteek Addanki, and Dekai Wu. From finite-state to inversion transductions: Toward unsupervised bilingual grammar induction. In *COLING-2012*, pp. 2325–2340, Mumbai, Dec 2012.

Harold L. Somers. *Valency and Case in Computational Linguistics*, vol. 3 of *EDITS*. Edinburgh, 1987.

Luc Steels, ed. *Computational Issues in Fluid Construction Grammar*. Springer, 2012.

Lucien Tesnière. *Éleménts de syntaxe structurale*. Klincksieck, 2nd ed., 1969.

Dekai Wu. Stochastic Inversion Transduction Grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404, Sep 1997.

Dekai Wu. Alignment. In Nitin Indurkhya and Fred J. Damerau, eds., *Handbook of Natural Language Processing*, pp. 367–408. Chapman and Hall / CRC, second ed., 2010.

Dekai Wu. *Introduction to Text Alignment: Statistical Machine Translation Models from Bitexts to Bigrammars*. Springer, 2014. Forthcoming.

Richard Zens and Hermann Ney. A comparative study on reordering constraints in statistical machine translation. In *ACL-2003*, Sapporo, Aug 2003.