Computational Models of Morphological Copying

Hossep Dolatian and Jeffrey Heinz
hossep.dolatian,jeffrey.heinz@stonybrook.edu

Objectives & Introduction

Questions

1. What is morphological copying?
2. What are adequate computational models?
3. How can these models be learned from examples?

Contributions

1. Use of 2-way finite-state transducers (2-way FSTs)
2. Cross-linguistic database of copying morphemes (RedTyp)
3. Classification of 2-way FSTs
4. Learning algorithms for morphological copying

Morphological Copying / "Reduplication"

Languages can use either affixes **or copying** to mark meanings

- English plural: book → book-s
- Indonesian plural: buku → buku∼buku

Significant variation in copying processes (Rubino, 2013)

(1) Total reduplication = unbounded copy (83%)
C: wanita→wanita∼wanita
'woman'→'women' (Indonesian)

(2) Partial reduplication = bounded copy (75%)
a. CV: guyon→gu∼guyon
'to jest'→'to jest repeatedly' (Sundanese)
b. CVC: takki→ tak∼takki
'leg'→'legs' (Agta)
c. CVCV: banagaɲu→bana∼banagaɲu
'return' (Dyirbal)

(3) And more (Moravcsik, 1978; Rubino, 2005; Inkelas and Zoll, 2005; Raimy, 2011; Urbanczyk, 2007)

2-way Finite-State Transducers

2-way FSTs can reread parts of the input string unlike 1-way FSTs (Engelfriet and Hoogeboom, 2001; Shallit, 2009; Filiot and Reynier, 2016)

- 1-way FSTs ⊂ 2-way FSTs ⊂ Turing Machines
- 2-way FSTs ≡ MSO-definable string transductions ≡ Streaming String Transducers

Why 2-way FSTs?

1. 2-way FSTs can model every encountered reduplicative morpheme, including unbounded copying. 1-way FSTs cannot model unbounded copying

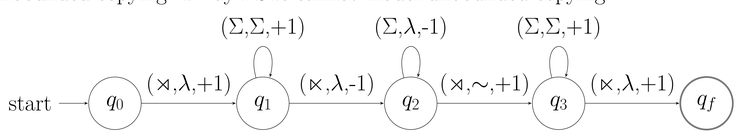

Figure 1: 2-way FST for total reduplication in Indonesian.

2. When modeling bounded copying, 1-way FSTs are orders of magnitude larger than 2-way FSTs.
3. 2-way FSTs directly capture linguistic generalizations of copying while 1-way FSTs do not.
4. Functions 2-way FSTs define are exactly the "regular functions," i.e. those definable with MSO logic (Engelfriedt and Higgeboom 2001).

RedTyp Database

RedTyp is an SQL database of reduplicative morphemes and 2-way FSTs for them.

- 138 reduplicative morphemes
- 93 languages from canonical surveys (Moravcsik, 1978; Rubino, 2005)
- 57 distinct 2-way FSTs models
- 8.8 is average FST size, measured in number of states
- Supplemental Python implementation for 2-way FSTs

Useful for:

- **Computational linguists:** Design 2-way FSTs to build morphological analyzers
- **Mathematical linguists:** Analyze 2-way FSTs to discover properties of reduplication
- **Machine Learning community:** Use the 2-way FSTs as learning targets
- **Other scientists:** Study 2-way FSTs for copying in other disciplines (biology, robotics)

Subclasses of 2-way FSTs

Subclasses of regular sets and functions have proved useful for studying phonology and morphology (Heinz, 2011a,b; Chandlee, 2014, 2017).

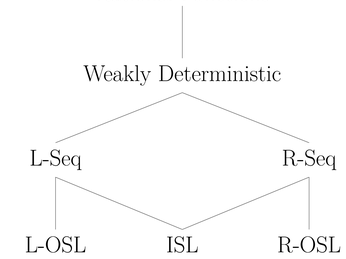

Figure 2: Sublcasses of functions definable with 1-way FSTs

Subclasses of 2-way FSTs

Defined with **concatenation of 1-way FSTs** and **composition**.

- **C-OSL :=** $\{f \mid f = g_1 \cdot g_2, g_1, g_2 \in \text{OSL}\}$
- **C-Seq :=** $\{f \mid f = g_1 \cdot g_2, g_1, g_2 \in \text{Seq}\}$
- **Com-C-OSL :=** $\{f \mid f = g_1 \circ g_2, g_1, g_2 \in \text{C-OSL}\}$
- **Com-C-Seq :=** $\{f \mid f = g_1 \circ g_2, g_1, g_2 \in \text{C-Seq}\}$

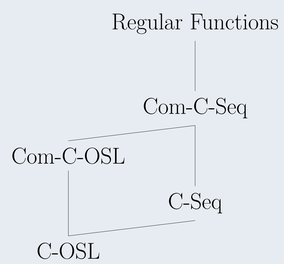

Figure 3: Sublcasses of functions definable with 2-way FSTs

Of the 138 reduplicative patterns in RedTyp ...

1. 102 are C-OSL
2. 30 are C-Seq
3. 4 are Com-C-OSL or Com-C-Seq (These involve phonological and morphological interactions)
4. 2 are regular (These involve phonological and morphological interactions)

Inferring C-OSL functions

Target Functions: $f(x) = g_1(x) \cdot g_2(x)$ where g_i are OSL functions.
Input Data D: input and output forms with the ∼ boundary
Example: $D = \{$(cat, cat∼cat), (bird, bir∼bird), (music, mus∼music) ...$\}$

Algorithm (sketch) for learning C-OSL functions:

1. For each input-output pair (input,copy$_1$∼copy$_2$) in D:
   - $s_1 \leftarrow s_1 \cup \{$(input, copy$_1$)$\}$
   - $s_2 \leftarrow s_2 \cup \{$(input, copy$_2$)$\}$
2. Submit s_1 and s_2 to the OSL learner (Chandlee et al., 2015) which returns transducers recognizing functions g_1 and g_2, respectively.
3. Return the 2-way FST recognizing $f = g_1 \cdot g_2$.

Next Steps

- Above learner exploits existence of morpheme boundaries
- We have shown with the boundary that learning morphological copying can be largely reduced to learning OSL functions, itself a solved problem (Chandlee et al., 2015).
- Thus, without the boundary, learning is reduced to learning morpheme segmentations (an open problem).

Conclusions

- Reduplication can be modeled with 2-way FSTs
- 2-way FSTs are compact, convenient, and capture linguistic generalizations
- 2-way FSTs opens doors for:
  1. Studying linguistic variation with the RedTyp database
  2. Identifying computationally natural subclasses that provide insight into linguistic variation
  3. Learning reduplication via inducing 2-way FSTs

References

Chandlee, J. (2014). Strictly Local Phonological Processes. Ph. D. thesis, University of Delaware.

Chandlee, J. (2017). Computational locality in morphological maps. Morphology, 1–43.

Chandlee, J., R. Eyraud, and J. Heinz (2015, July). Output strictly local functions. In Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015), Chicago, USA, pp. 112–125.

Engelfriet, J. and H. J. Hoogeboom (2001). Mso definable string transductions and two-way finite-state transducers. ACM Transactions on Computational Logic (TOCL) 2(2), 216–254.

Filiot, E. and P.-A. Reynier (2016). Transducers, logic and algebra for functions of finite words. ACM SIGLOG News 3(3), 4–19.

Heinz, J. (2011a). Computational phonology part I: Foundations. Language and Linguistics Compass 5(4), 140–152.

Heinz, J. (2011b). Computational phonology part II: Grammars, learning, and the future. Language and Linguistics Compass 5(4), 153–168.

Inkelas, S. and C. Zoll (2005). Reduplication: Doubling in Morphology. Cambridge University Press.

Moravcsik, E. (1978). Reduplicative constructions. In J. Greenberg (Ed.), Universals of Human Language, Volume 1, pp. 297–334. Stanford, California: Stanford University Press.

Raimy, E. (2011). Reduplication. In M. van Oostendorp, C. Ewen, B. Hume, and K. Rice (Eds.), The Blackwell Companion to Phonology. Blackwell.

Rubino, C. (2005). Reduplication: Form, function and distribution. pp. 11–29. Berlin: Mouton de Gruyter.

Rubino, C. (2013). Reduplication. Leipzig: Max Planck Institute for Evolutionary Anthropology.

Shallit, J. (2009). A second course in formal languages and automata theory, Volume 179. Cambridge University Press Cambridge.

Urbanczyk, S. (2007). Themes in phonology. The Cambridge Handbook of Phonology, edited by Paul de Lacy, 473–493.