Learning Transfer Rules without Templates

Michael Wayne Goodman

University of Washington

August 8, 2017

My work in MT has shifted somewhat:

My work in MT has shifted somewhat:

- fewer exciting, novel approaches to deep semantic transfer,

My work in MT has shifted somewhat:

- fewer exciting, novel approaches to deep semantic transfer,
- but more expansions of the traditional JaEn transfer-based system,

My work in MT has shifted somewhat:

- fewer exciting, novel approaches to deep semantic transfer,
- but more expansions of the traditional JaEn transfer-based system,
- a bit less optimism,

My work in MT has shifted somewhat:

- fewer exciting, novel approaches to deep semantic transfer,
- but more expansions of the traditional JaEn transfer-based system,
- a bit less optimism,
- but more knowledge and advice for others,

My work in MT has shifted somewhat:

- fewer exciting, novel approaches to deep semantic transfer,
- but more expansions of the traditional JaEn transfer-based system,
- a bit less optimism,
- but more knowledge and advice for others,
- and some useful artifacts produced

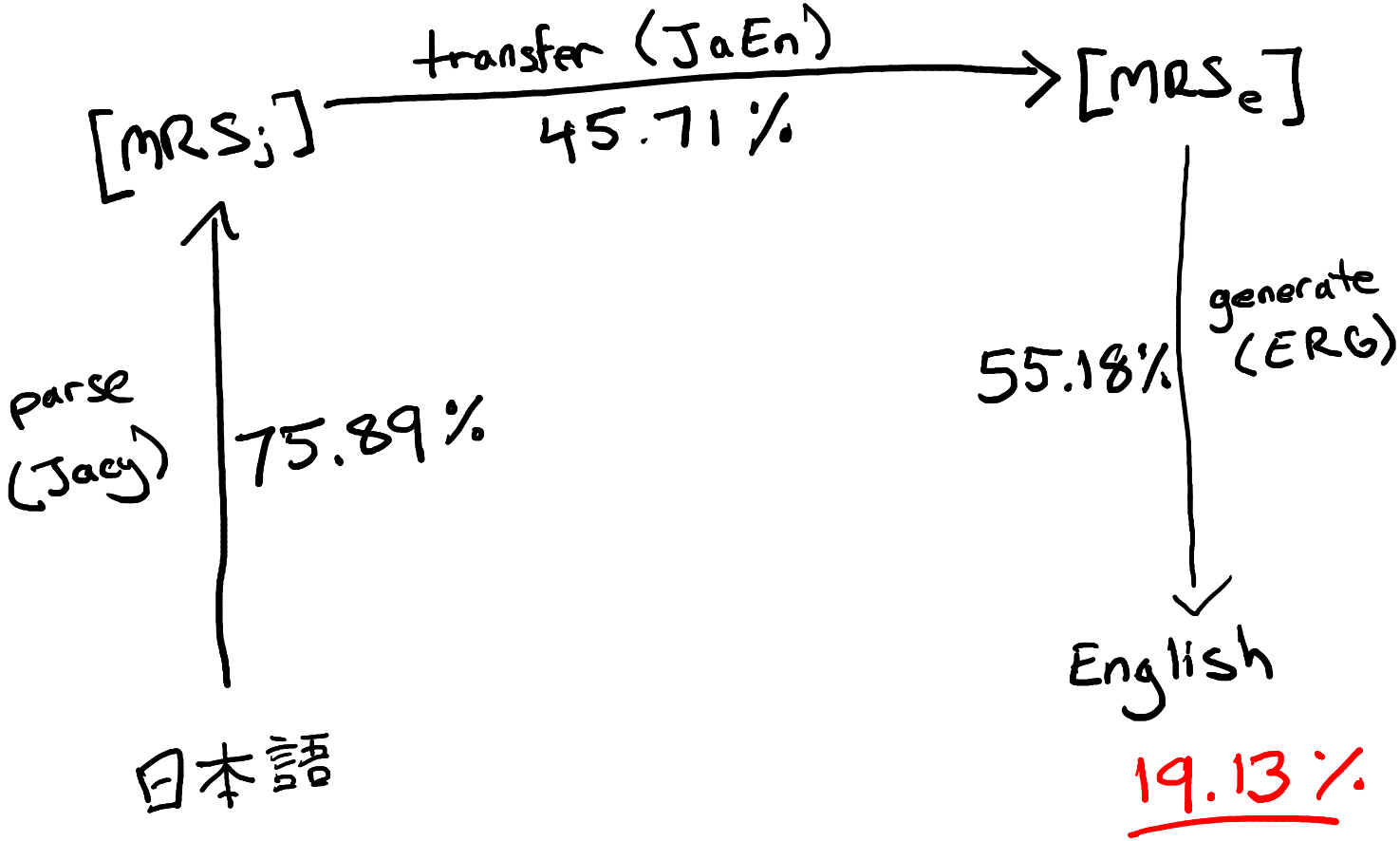
Recent work

- Established JaEn baseline
- Improved Jacy
- ACE-based translation environment
- Extracting rules

Reanimating JaEn

- Previous home: `${LOGONROOT}/uio/tm/jaen`
- New home: `https://github.com/delph-in/JaEn`
- Updated to work with ACE and the LKB
- Various bugs fixed
- Updated Petter's `select-rule.py` script
- Includes transfer rules extracted from Haugereid and Bond (2012) (maybe all of them?)
- Readme with setup instructions, citation.bib

Jacy + JaEn + ERG translation pipeline coverage
(relative coverage in black; absolute in red)



BLEU (NIST) scores for MT with JaEn using ACE:

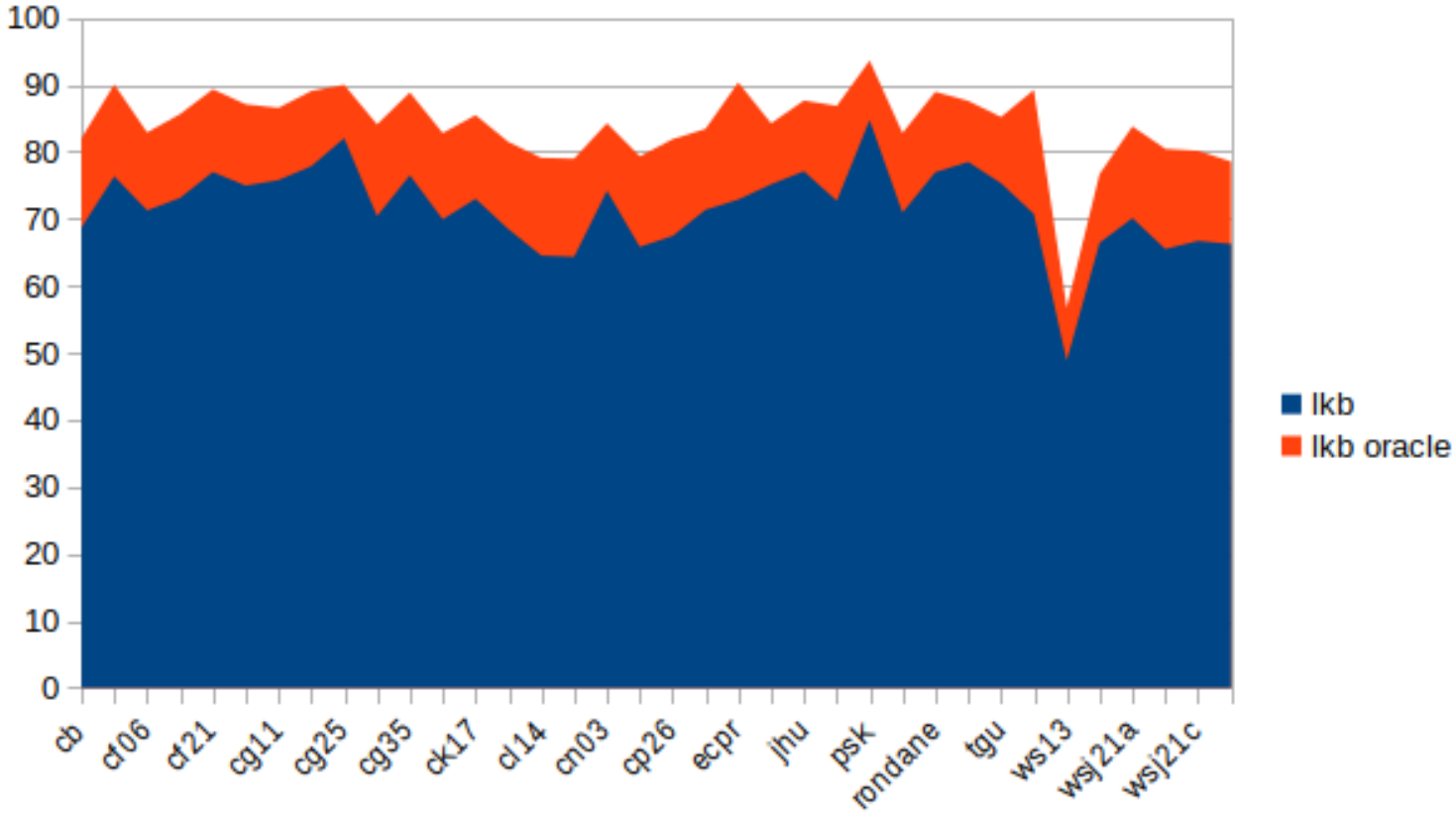
top 1	**7.26**
oracle (100)	**22.43**

This is just for the 19.13% items that survived to a translation!

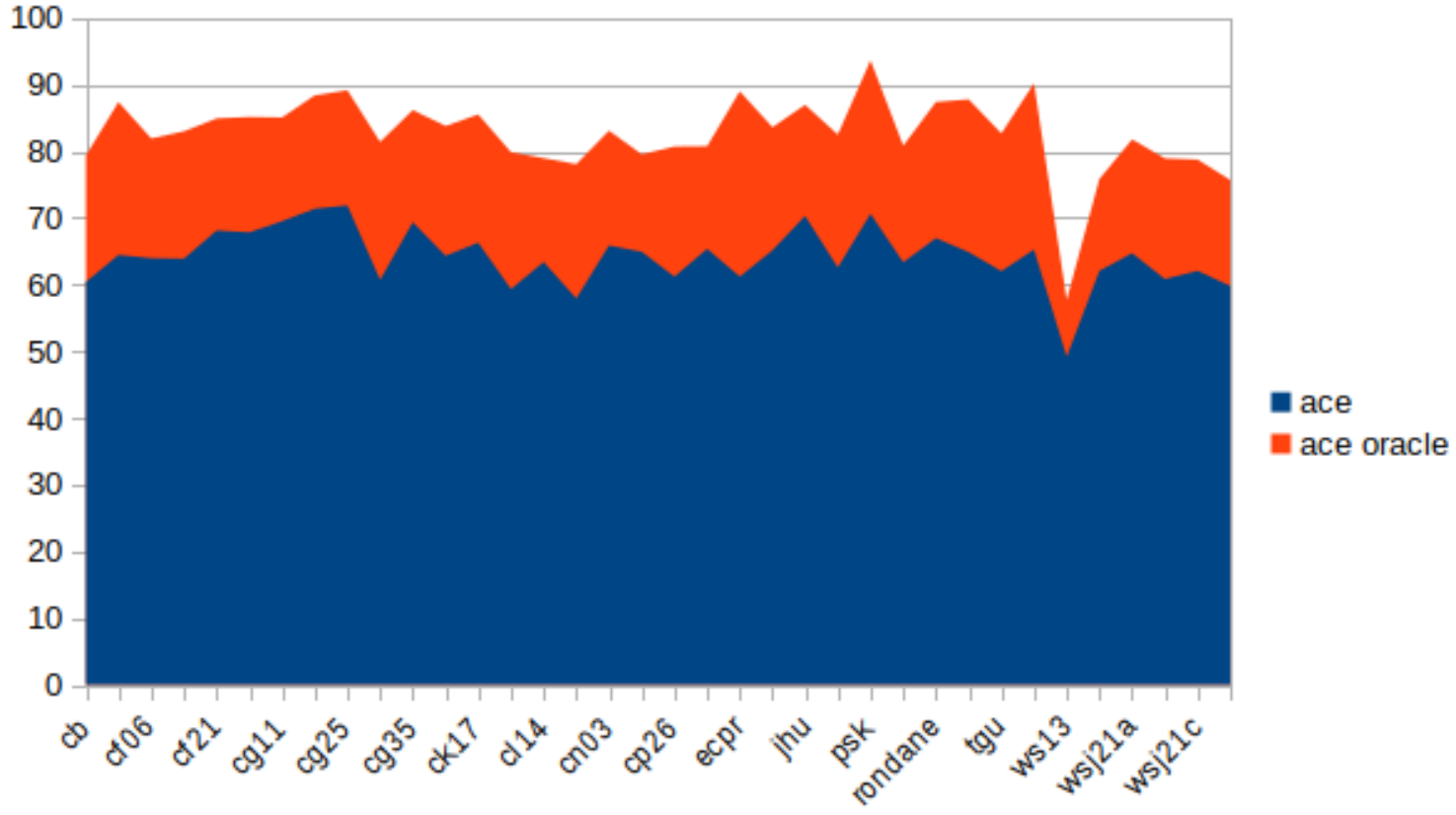
Compare to 2011 result (for 100 sentences):

	BLEU	**METEOR**	**Human**
JaEn	10.07	35.51	52.75
Moses	23.85	51.65	47.25

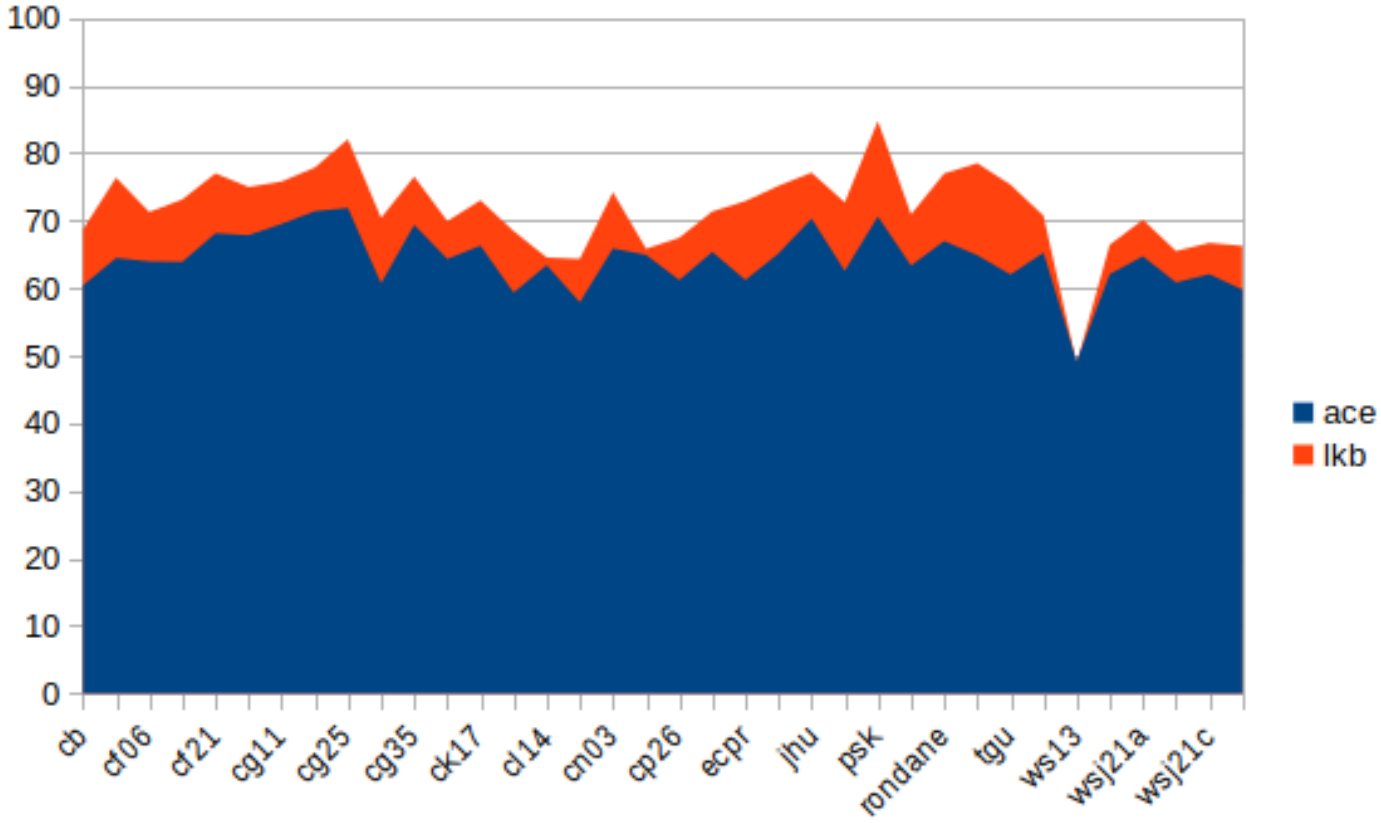
Detour: ERG generation performance, LKB vs ACE



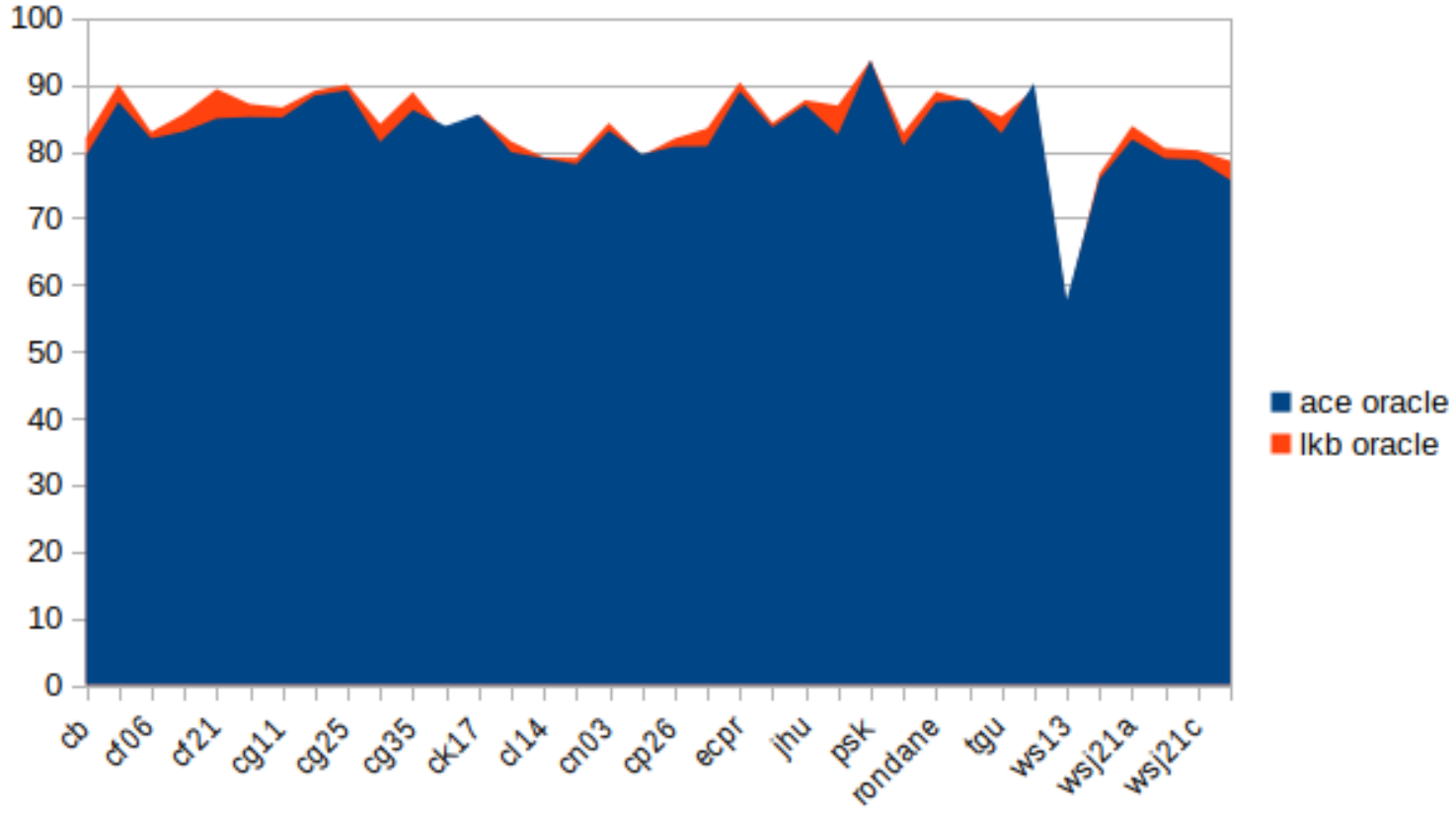
Detour: ERG generation performance, LKB vs ACE



Detour: ERG generation performance, LKB vs ACE



Detour: ERG generation performance, LKB vs ACE



Other sources of problems:

- Bad MRSs output from Jacy
- Old parse selection model for Jacy
- Divergence between JaEn, Jacy, and the ERG
- Generating from generic lexical entries
  - `_buckwheat_n_0`
  - `_porcelain/NN_u_unknown`
  - `[named<4:12> LBL: h7 CARG: "Takayuki"...]`

Lessons from working with JaEn

- Oracle BLEU isn't bad, but good reranking is important!
- Coverage (for the whole pipeline) is pretty terrible
- I probably couldn't have gotten it working at all without direct help from Francis

XMT (new tool)

- Manages processing tasks in a single [incr tsdb()]-like profile
- Each task gets its own ID
- Parsing (p-id)
- Transfer (x-id)
- Generating (g-id)
- Paraphrasing (r-id)

XMT

- Meant for the above tasks, not for grammar development
- Currently bundles some scripts for transfer rule extraction (predicate linearization, subgraph extraction, etc.)
- Development led to some PyDelphin improvements:
  - AceTransferer
  - More robust ACE processing (e.g., automatic restarts, timeouts)

Extracting transfer rules

- From word-aligned predicate strings
- From aligned subgraphs

Extracting transfer rules

- From word-aligned predicate strings
- From aligned subgraphs

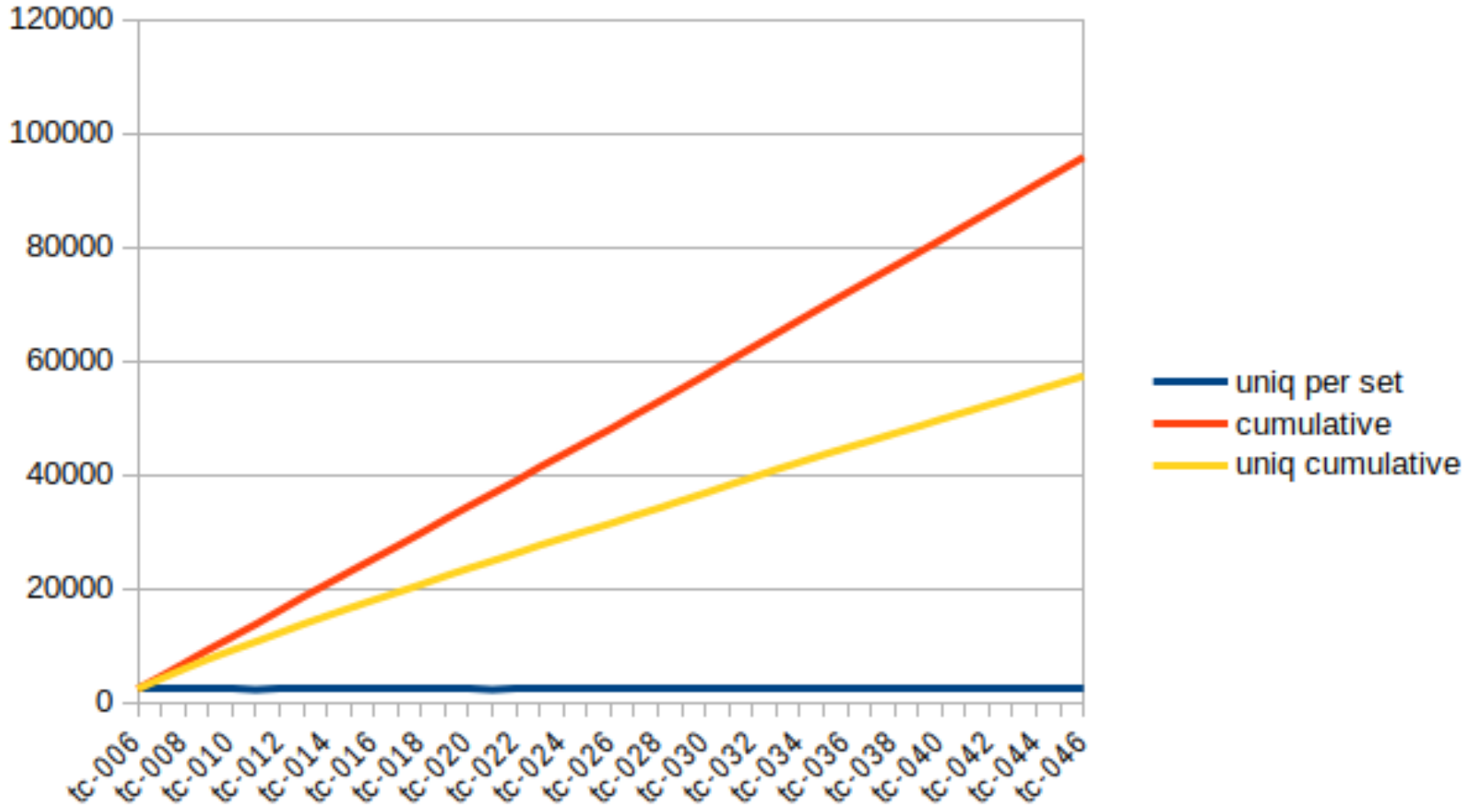
Extracting from aligned predicate strings

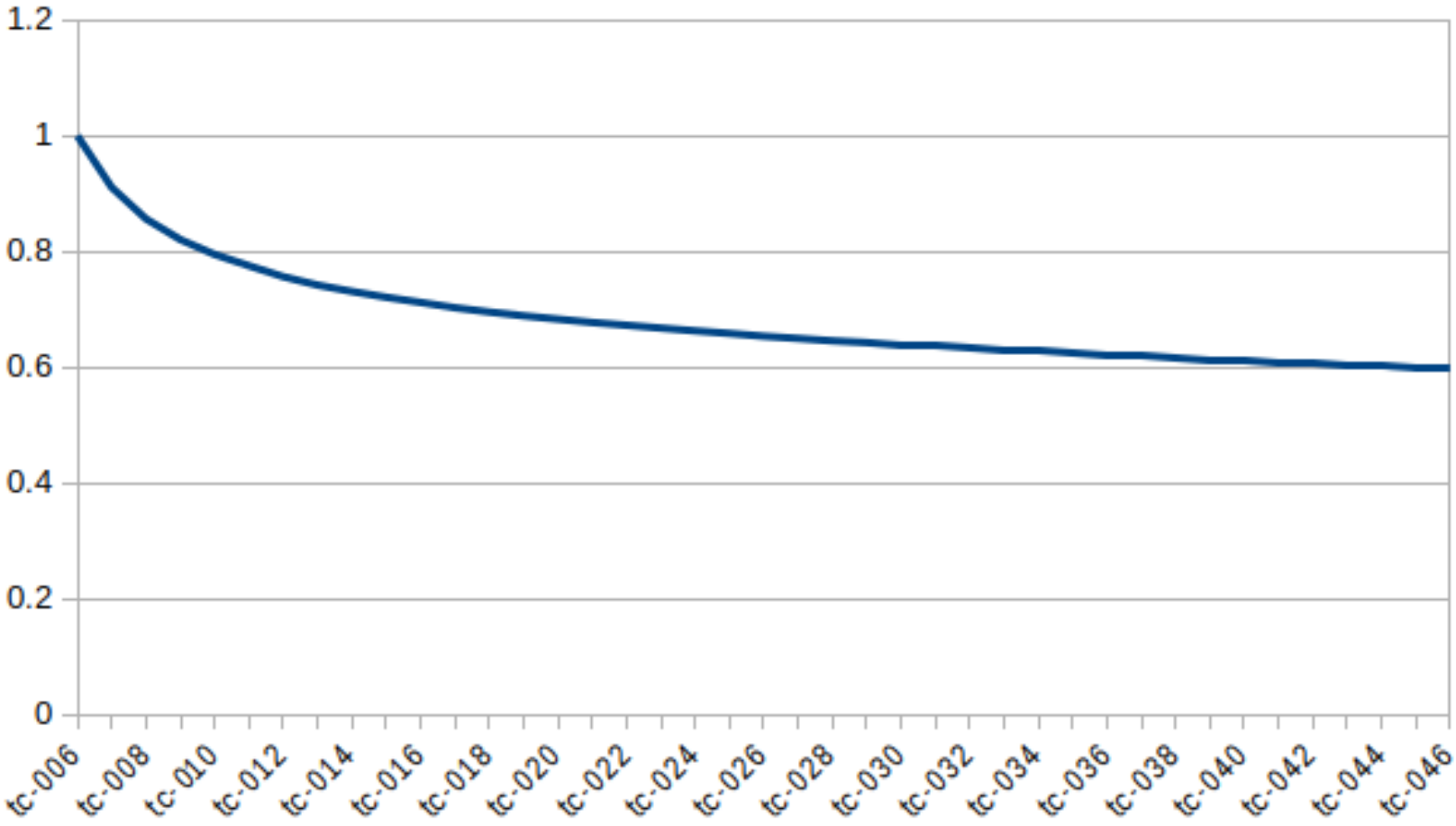
- for each EP sorted by CFROM, -CTO, output predicate
- done for source and target, we get new "bitext"
- get phrase alignments from, e.g., anymalign or giza++
- take aligned predicate phrases back to MRS graph
- extract source/target subgraphs from predicate phrases
  - does it match a template?
  - does the subgraph have other properties?
- use subgraphs to create new transfer rules

- predicate phrases (n-grams) are sensitive to adjacent context
- we can improve the alignment quality and increase quantity of useful phrases by blocking non-useful but predictable predicates
- drop `udef_q`, `pronoun_q`, `number_q`, `proper_q`, `def_q`
- `_wa_d`, `parg_d`, ...
- `compound`, `subord`, all abstract predicates?

Other filters

- source and target graphs are connected
- constrain top variable type
- maximum graph depth
- source/target graph size ratio
- minimum lexical weight
- minimum translation probability


120000
100000
80000
60000
40000
20000
0
tc-006
tc-008
tc-010
tc-012
tc-014
tc-016
tc-018
tc-020
tc-022
tc-024
tc-026
tc-028
tc-030
tc-032
tc-034
tc-036
tc-038
tc-040
tc-042
tc-044
tc-046
uniq per set
cumulative
uniq cumulative


1.2
1
0.8
0.6
0.4
0.2
0
tc-006
tc-008
tc-010
tc-012
tc-014
tc-016
tc-018
tc-020
tc-022
tc-024
tc-026
tc-028
tc-030
tc-032
tc-034
tc-036
tc-038
tc-040
tc-042
tc-044
tc-046

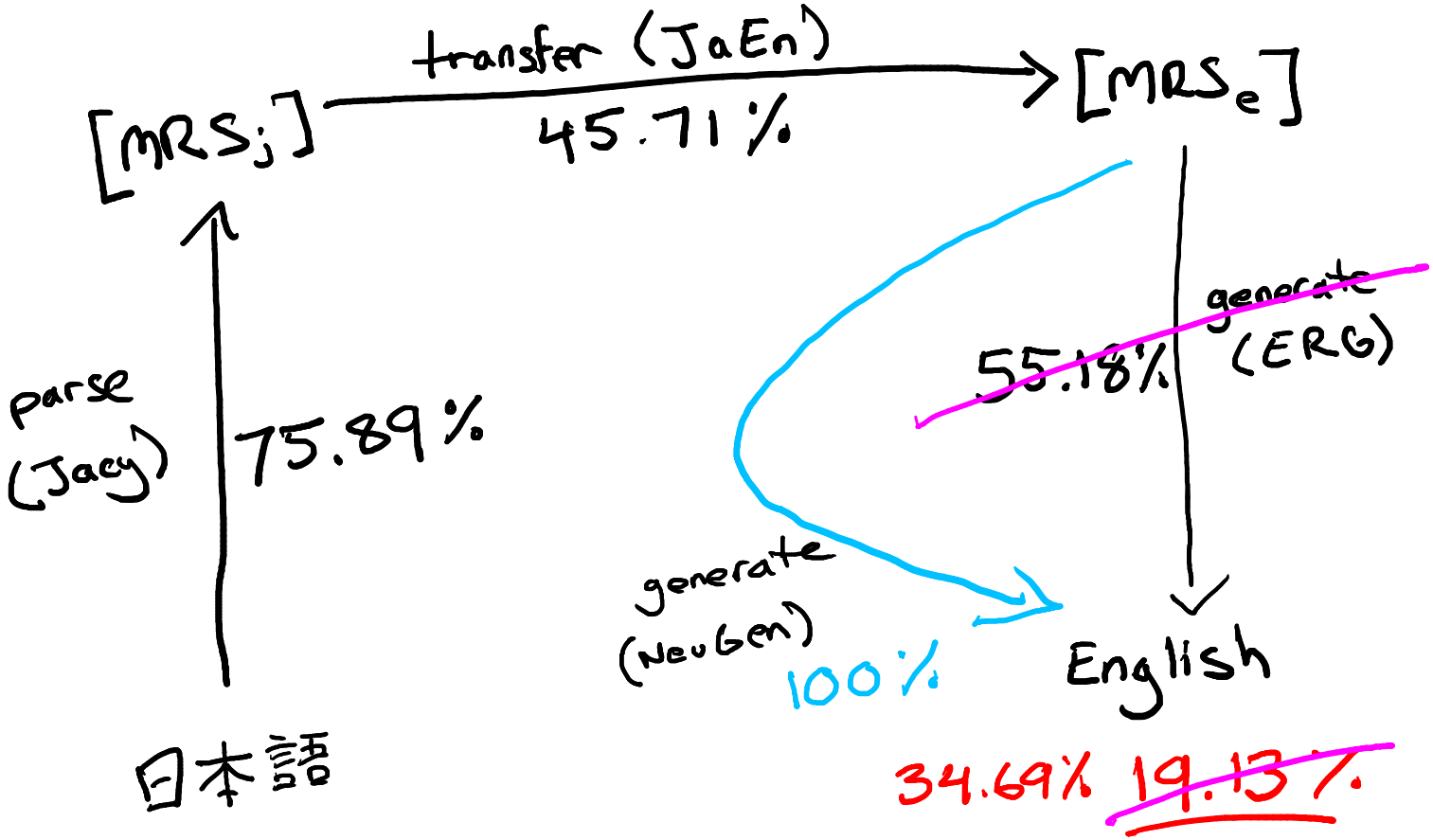
Second method, without aligned predicate phrases

- enumerate all subgraphs
- hypothesize every source/target pair (cartesian predicate) is a translation
- filter much as before
- let a function of the pair frequencies select the good translations
- should give more results, but at possibly lower quality, than previous method

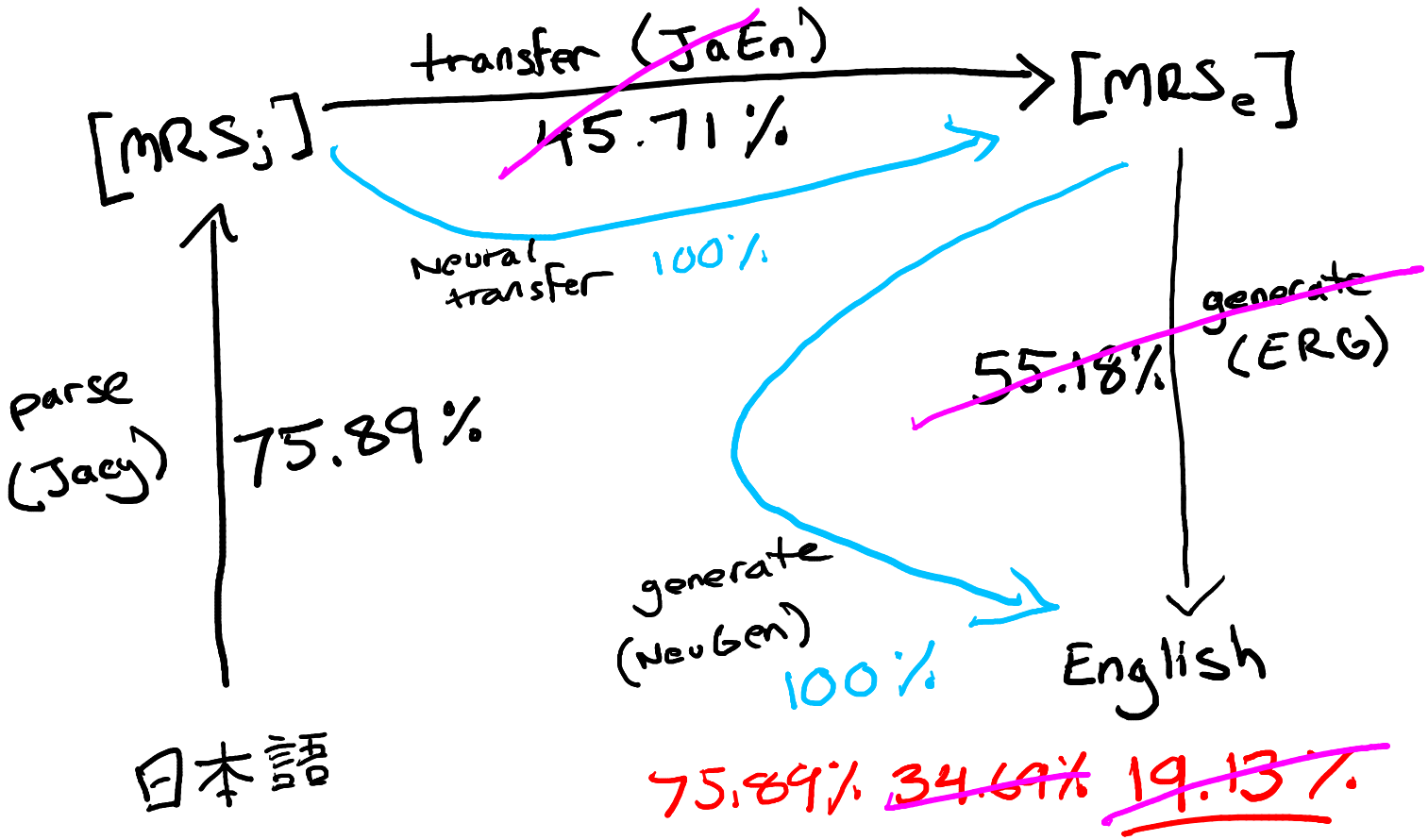
Third method (probably future work)

- learn weights for a graph grammar (e.g., (Groschwitz et al., 2015; Gilroy et al., 2017; Chiang et al., 2013))
- enumerate subgraphs as in method 2
- use graph composition score for ranking

Neural Generation (future work?)



Neural Transfer (future work?)



Advice and suggestions welcome!

Thanks

David Chiang, Jacob Andreas, Daniel Bauer, Karl Moritz Hermann, Bevan Jones, and Kevin Knight. 2013. Parsing graphs with hyperedge replacement grammars. In *ACL (1)*, pages 924–932.

Sorcha Gilroy, Adam Lopez, and Sebastian Maneth. 2017. Parsing graphs with regular graph grammars. *c 2017 The Association for Computational Linguistics*, page 199.

Jonas Groschwitz, Alexander Koller, Christoph Teichmann, et al. 2015. Graph parsing with s-graph grammars. In *ACL (1)*, pages 1481–1490.

Petter Haugereid and Francis Bond. 2012. Extracting semantic transfer rules from parallel corpora with smt phrase aligners. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 67–75. Association for Computational Linguistics, Jeju, Republic of Korea. URL `http://www.aclweb.org/anthology/W12-4208`.