GENERALIZED OPTIMAL LINEAR ORDERS

M.S.THESIS

RISHI BOMMASANI

CORNELL UNIVERSITY

MOTIVATION

Goals:

- Unified theory for linear word order
- Improved downstream NLP systems

- Human representations are not machine-optimal
  - language, gestures, mathematics, visual communication vs. bits

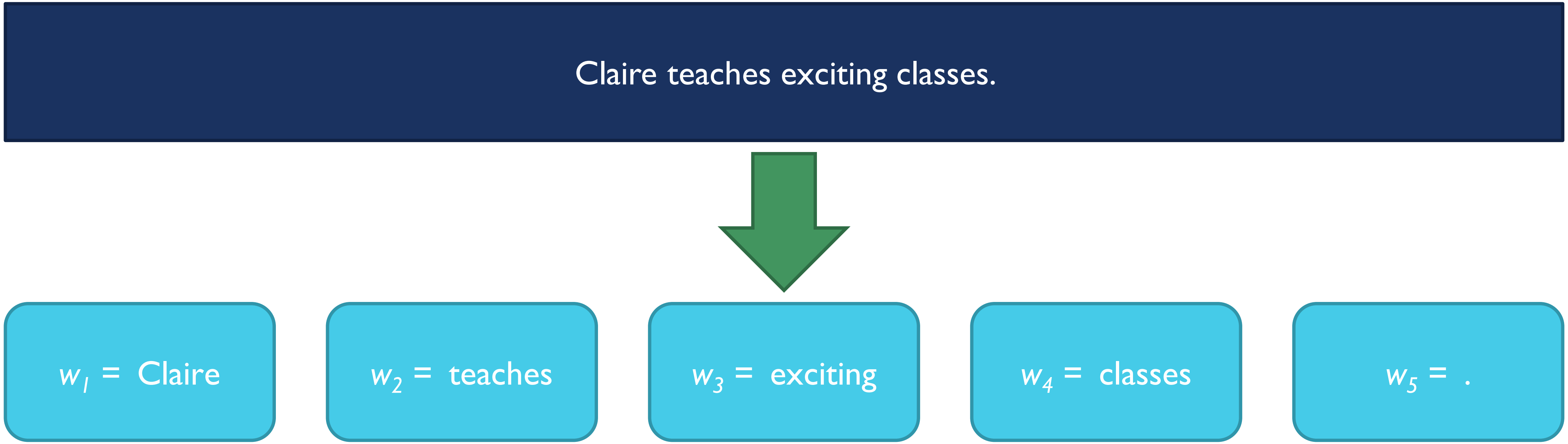
- Human representations in language are not machine-optimal
  - characters, words, sentences vs. numbers, vectors, linear transformations

- Word order is a fundamental property of language
  - Word order is (generally) how we convey syntax
  - We are pretty good at a broad class of lexical/word-level representation learning

- Combinatorial space of possibilities
  - Tremendously unexplored in NLP; O(1) orders normally considered
  - Natural algorithmic challenges for exploring insightfully

PRIMITIVES

- Unit of analysis: *sentence*
  - Representation: sequence of words
  - *Assumption:* gold-standard tokenization



Claire teaches exciting classes.

w_1 = Claire

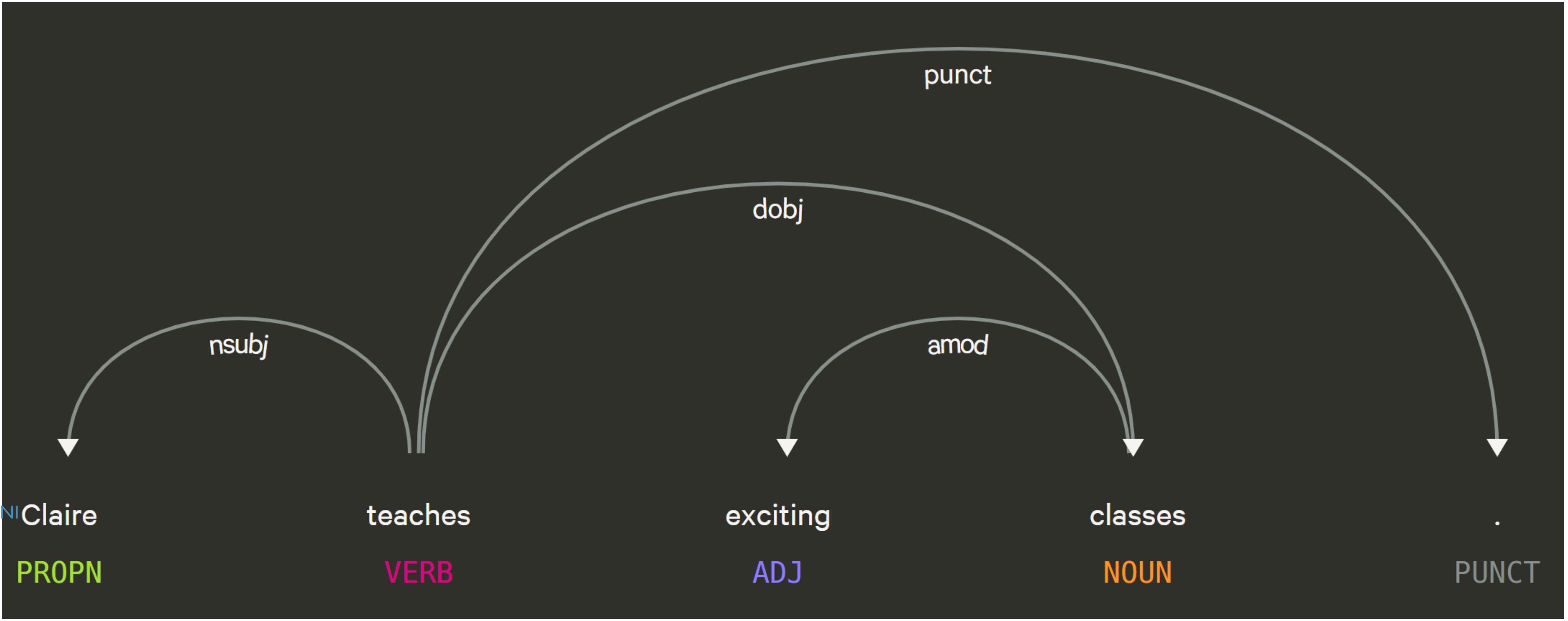
w_2 = teaches

w_3 = exciting

w_4 = classes

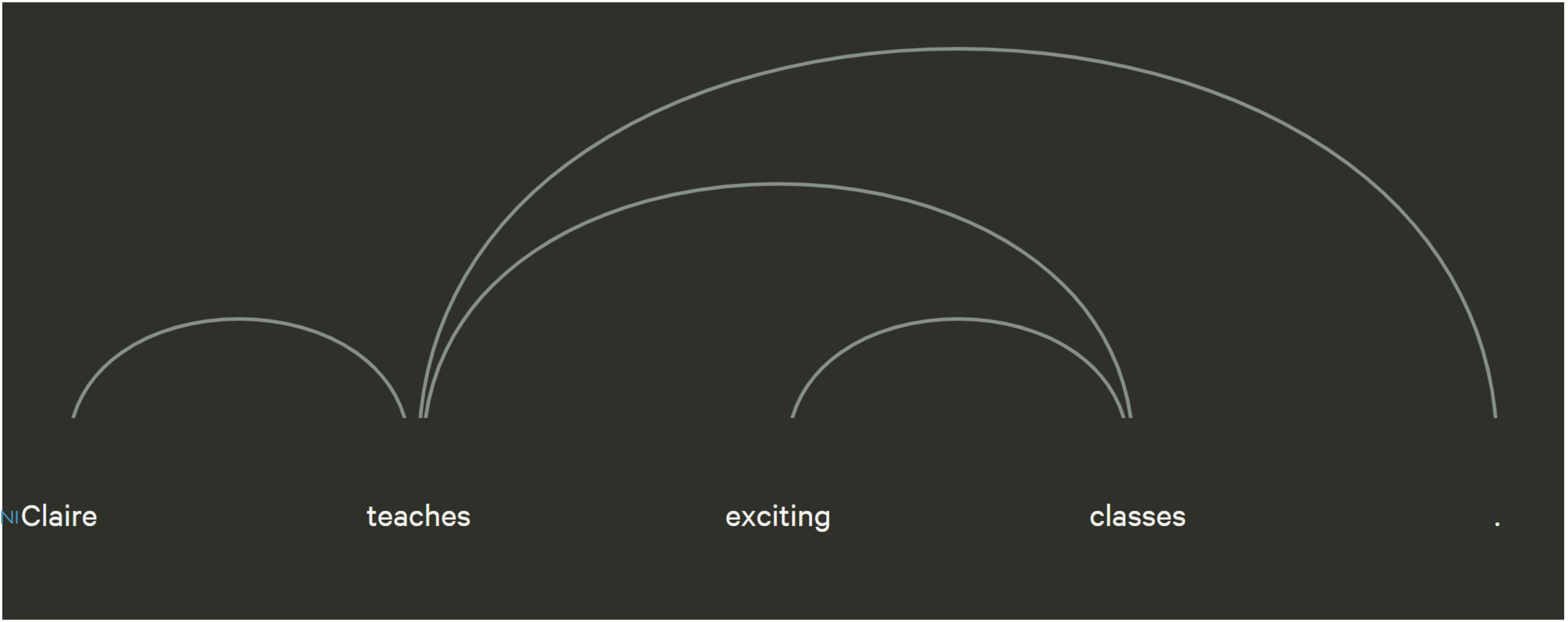
w_5 = .

- Scaffold: *dependency parse*
  - Representation: Graph $G = (V, E)$
  - $V = \{w_i \mid i \in [n]\}$, E = {directed, labelled binary dependency relations}



Scaffold: *dependency parse*

- Representation: Graph $G = (V, E)$
- $V = \{w_i \mid i \in [n]\}$, E = {~~directed, labelled~~ binary dependency relations}



Requirement: *dependency parser*

- *Assumption / Limitation*: High-quality dependency parser
- *Dependency parsing*: sentence → dependency parse

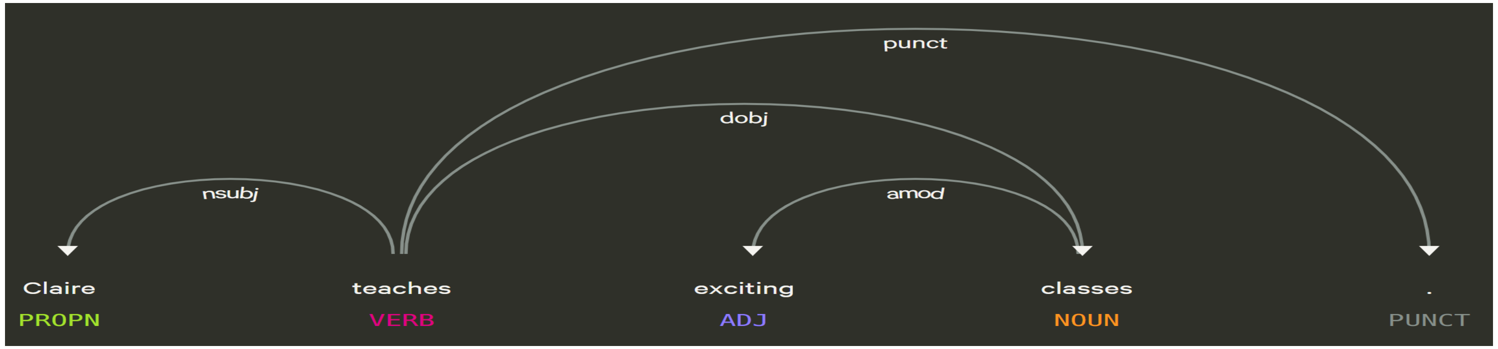
w_1 = Claire

w_2 = teaches

w_3 = exciting

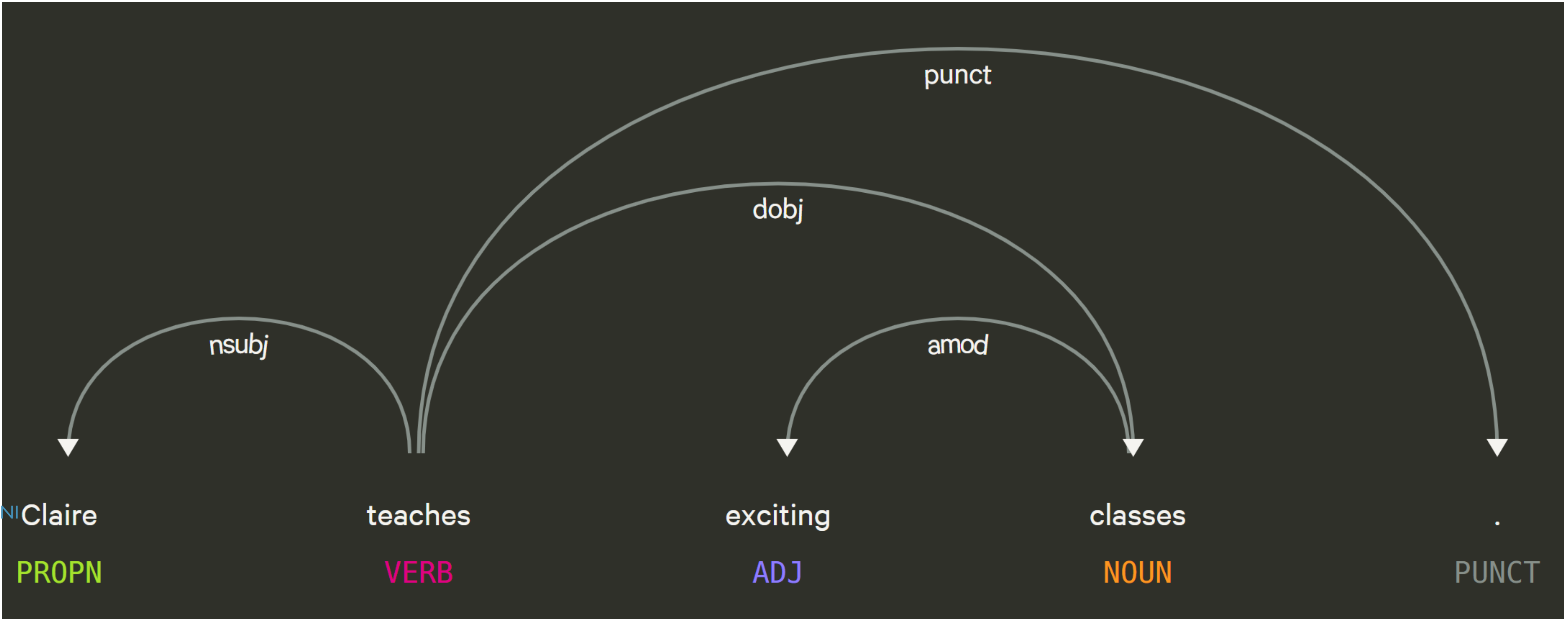
w_4 = classes

w_5 = .



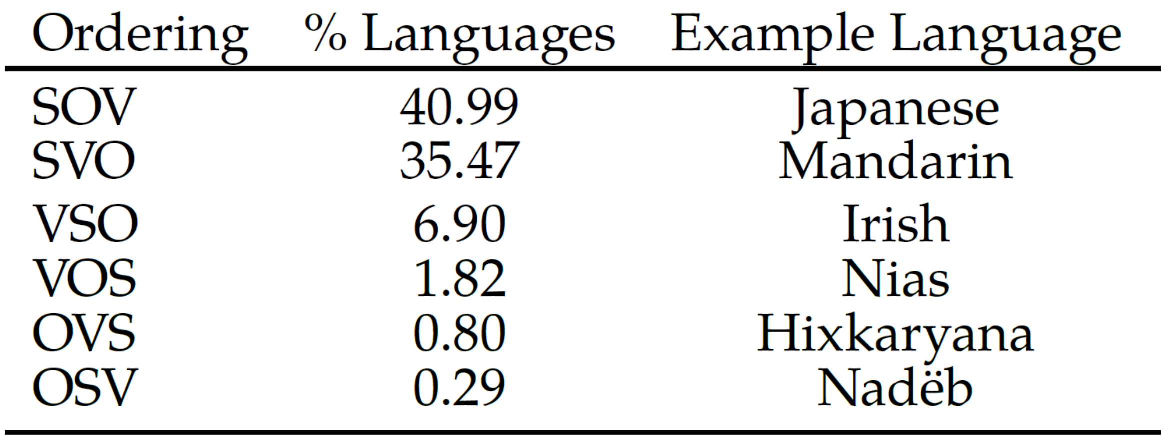
Property: *projectivity*

- We will draw dependency parses is this canonical way (all edges above sentences)
- *Projective:* The parse has no intersecting/crossing edges when drawn this way
  - Common property of interest in dependency parsing, well-studied linguistically



WORD ORDER AND HUMAN LANGUAGE

- Word ordering behaviors
  - Rigid (Fixed) vs. Flexible (Free)
  - Basic word order



Ordering	% Languages	Example Language
SOV	40.99	Japanese
SVO	35.47	Mandarin
VSO	6.90	Irish
VOS	1.82	Nias
OVS	0.80	Hixkaryana
OSV	0.29	Nadëb

- Why does this matter?
  - Language universals (Greenberg, 1963)
    - Harmonic orders facilitate learning (Culbertson and Newport, 2015, 2017)
  - Typological categorization of natural languages (Dryer, 1997)
    - *WALS* (Dryer, 2013)

Incremental Processing Theories

- Goal: Explain processing cost of w_i given context ($w_i \ldots w_{i-1}$)

- Expectation-based theories:
  - Cost is proportional to $\log \frac{1}{p(w_i \mid w_1 \ldots w_{i-1})}$ (Hale, 2001; Levy, 2008)
- Memory-based theories:
  - Cost is proportional to difficulty of retrieving units used to process w_i (Gibson, 1998)

Dependency Length Minimization

- *What belongs together mentally is placed together* (Behaghel, 1932)
- Pervasive evidence:
  - Japanese – (Yamashita and Chang 2001); 37 natural languages – (Futrell et al. 2015)
  - Grammar design (Rijkhoff, 1990; Hawkins, 1990)
  - Beneficial inductive bias in parsing (Collins, 2003, Klein and Manning, 2004; Eisner and Smith, 2005; Smith and Eisner, 2006)

a. Bob **threw out** the trash.

b. Bob **threw** the trash **out**.

c. Bob **threw out** the old trash that had been sitting in the kitchen for several days.

d. Bob **threw** the old trash that had been sitting in the kitchen for several days **out**.

ALGORITHMIC FRAMEWORK

PART I: NOTATION AND OBJECTIVES

Inputs: Sentence s = (w_l … w_n); dependency parse $G = (V, E)$

Linear layout – A bijective mapping $\pi : V \to [n]$

Identity linear layout – $\pi_I : V \to [n]$

$$\pi_I(w_i) = i$$

Edge length – $d_\pi : E \to \mathbb{N}$

$$d_\pi(u, v) = |\pi(u) - \pi(v)|$$

$$L_\pi(i) = \{ u \in V : \pi(u) \leq i\};\ R_\pi(i) = \{ u \in V : \pi(u) > i\}$$

Edge cut – $\theta_\pi : [n] \to \mathbb{N}$

$$\theta_\pi(i) = |\{(u, v) \in E : u \in L_\pi(i) \text{ and v} \in R_\pi(i)\}|$$

Let D be the space of input sentences and S_n be the space of linear layouts on s.

Ordering rule – $r : D \rightarrow S_n$

Number of ordering rules: $(n!)^{|D|}$

Compute ordering rules via optimization *wrt* cost function f:

$$r_f(s) = \operatorname*{argmin}_{\pi \in S_n} f(\pi, s)$$

	Bandwidth	Minimum Linear Arrangement (MinLA)	Cutwidth	sum-Cutwidth
Objective	$\max_{(u,v) \in E} d_{\pi}(u,v)$	$\sum_{(u,v) \in E} d_{\pi}(u,v)$	$\max_{i \in [n]} \theta_{\pi}(i)$	$\sum_{i \in [n]} \theta_{\pi}(i)$
Interpretation	Minimize longest dependency	Minimize sum of length of dependencies	Minimize max # of active dependencies	Minimize sum of active dependencies
Abbreviations	r_b, π_b	r_m, π_m	r_c, π_c	
Origin	Harary (1967)	Harper (1964)	Adolphson and Hu (1973)	
Graphs	NP-Hard Papadimitriou (1976)	NP-Hard Garey et al. (1974)	NP-Hard Gavril (2011)	
Tree	NP-Hard Garey et al. (1978)	$n^{1.58}$ Chung (1984)	$n \log n$ Yannakakis (1985)	

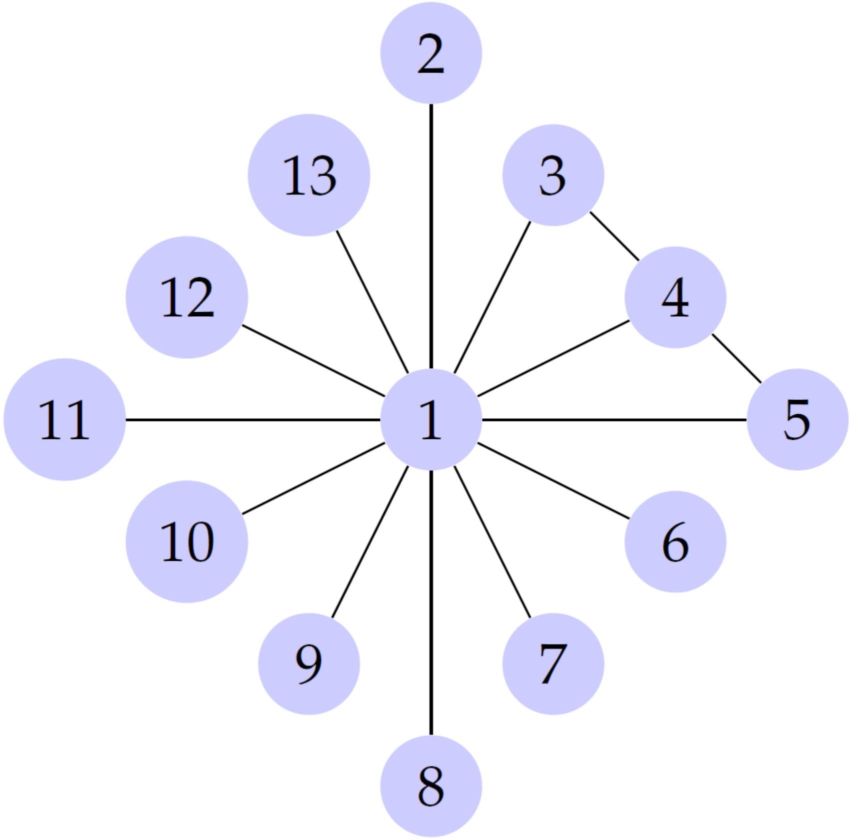
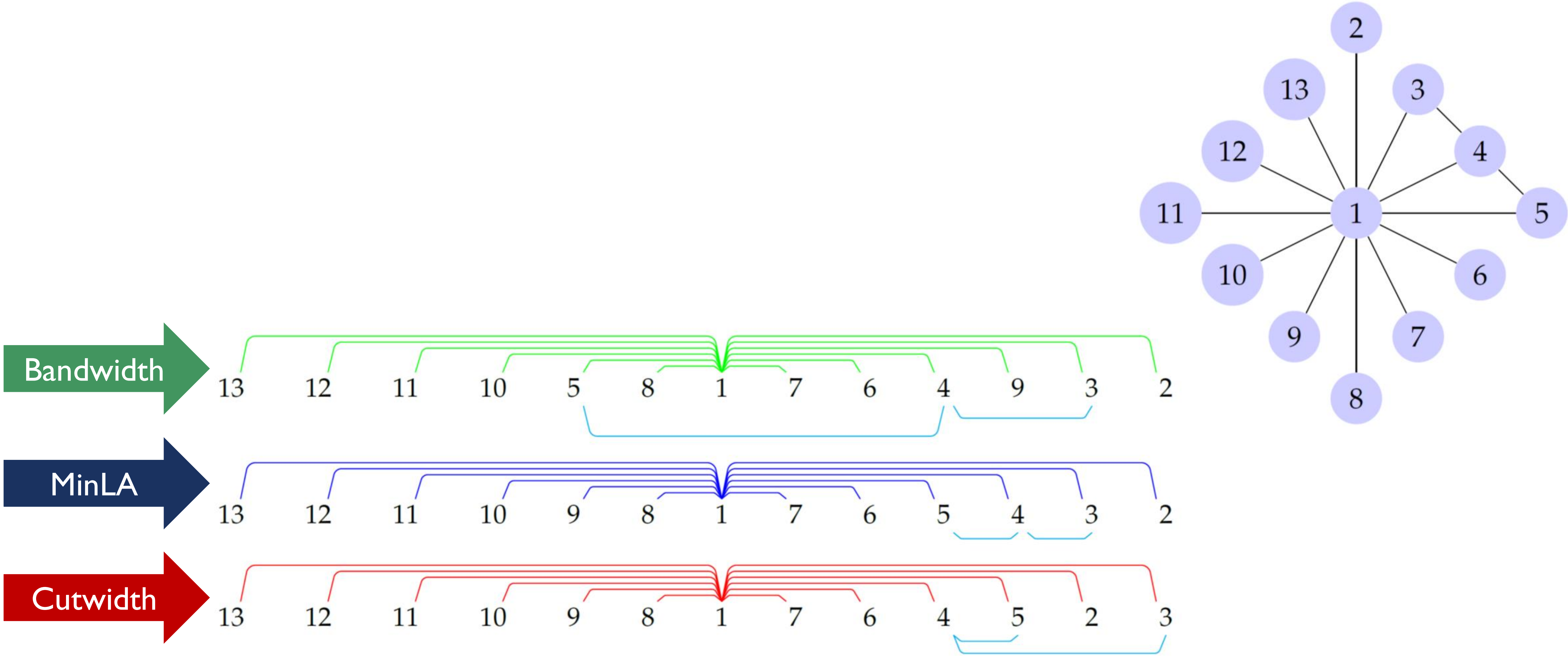


Figure 4.2: A graph $\mathcal{G}$ with a linear layout specified by the vertex labels in the figure. Given this linear layout, the bandwidth is 12 (this is $13 - 1$), the cutwidth is 12 (this is due to position 1), and the minimum linear arrangement score is 80 $\left(\text{this is } \sum_{i=2}^{13}(i-1) + (4-3) + (5-4)\right)$.



Bandwidth

MinLA

Cutwidth

Figure 4.3: Solutions for optimizing each of the three objectives for the graph given in Figure 4.2. The linear layout is conveyed via the linear ordering and the numbers refer to the original vertices in the graph (as shown in Figure 4.2). The top/green graph is bandwidth-optimal (bandwidth of 6), the middle/blue graph is minimum linear arrangement-optimal (minimum linear arrangement score of 44), the bottom/red graph cutwidth-optimal (cutwidth of 6). The cyan edges drawn below the linear sequence convey the difference in the optimal solutions.

- Generalizing Bandwidth and MinLA

$$r_p(\pi, \bar{s}) = \arg\min_{\pi \in S_n} \left(\sum_{(w_i, w_j) \in \mathcal{E}} d_\pi(w_i, w_j)^p \right)^{1/p}$$

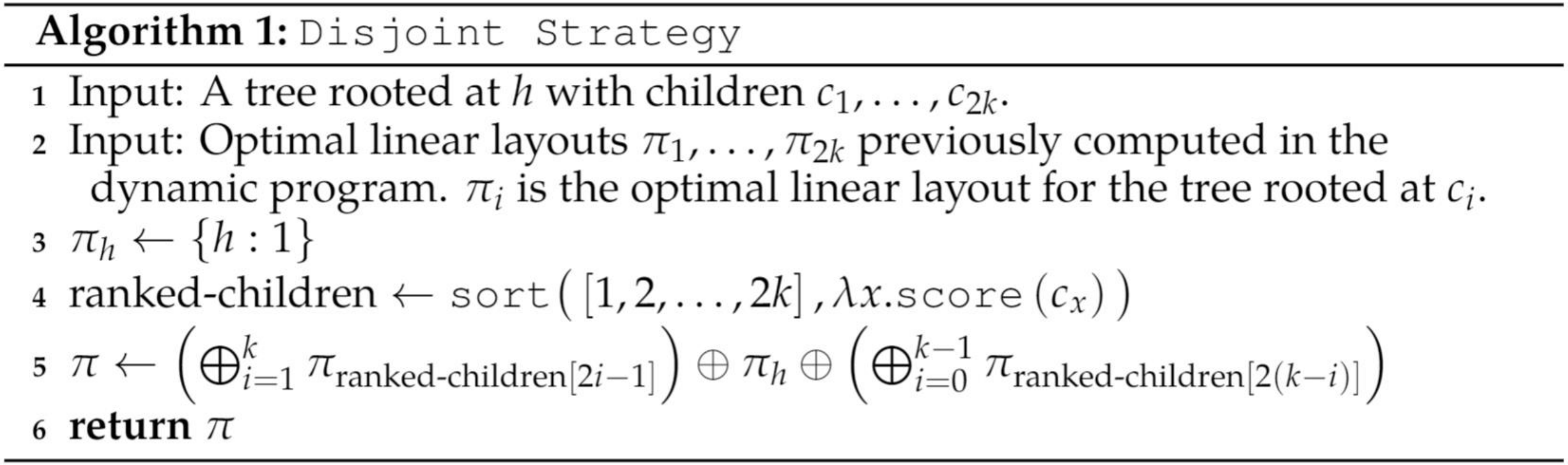
- Equating MinLA and sum-Cutwidth
  - MinLA : Each edge contributes it length
  - sum-Cutwidth: Each edge contributes 1 to every position from its left endpoint up until its right endpoint.

ALGORITHMIC FRAMEWORK

PART II: ALGORITHMS

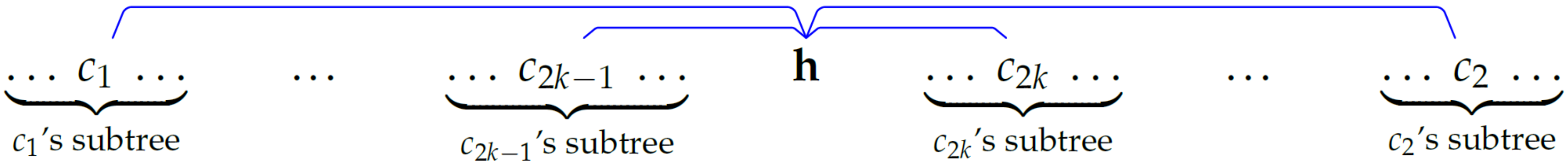
- Bandwidth
  - Heuristic from Cuthill and McKee (1969); modified by Chan and George, 1980
  - Breadth-first search in order of increasing degree
- MinLA
  - Linear time *under projectivity constraints** from Gildea and Temperley, 2007
  - Will correct some errors and show it uses the same ideas as Yannakakis, 1985
- Cutwidth
  - Linear time *under projectivity constraints** from Yannakakis, 1985
  - Dynamic programming over tree structure known as *disjoint strategy*

* In the algorithms community, this is sometimes referred to the PLANAR version of the problem (e.g. PLANAR-CUTWIDTH)



Algorithm 1: `Disjoint Strategy`

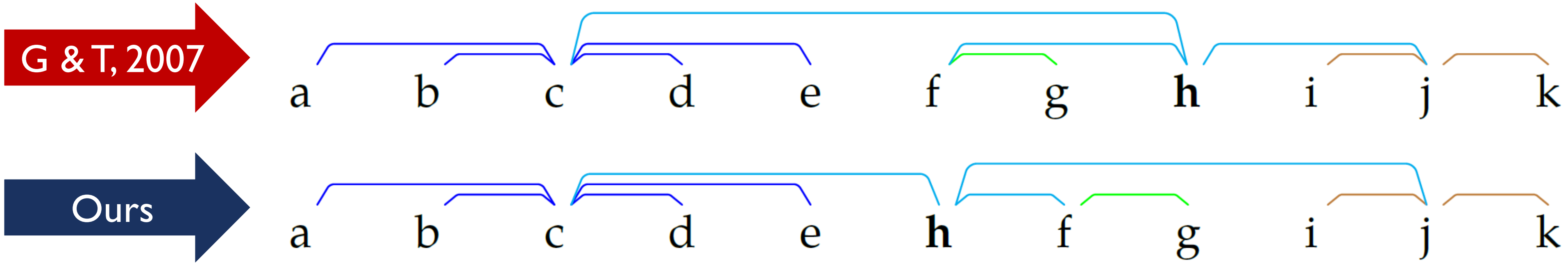
```
Input: A tree rooted at $h$ with children $c_1, \ldots, c_{2k}$.
Input: Optimal linear layouts $\pi_1, \ldots, \pi_{2k}$ previously computed in the
  dynamic program. $\pi_i$ is the optimal linear layout for the tree rooted at $c_i$.
$\pi_h \leftarrow \{h : 1\}$
ranked-children $\leftarrow$ sort([1,2,...,2k], $\lambda x$.score($c_x$))
$\pi \leftarrow \left(\bigoplus_{i=1}^{k} \pi_{\text{ranked-children}[2i-1]}\right) \oplus \pi_h \oplus \left(\bigoplus_{i=0}^{k-1} \pi_{\text{ranked-children}[2(k-i)]}\right)$
return $\pi$
```



Correcting Gildea and Temperley, 2007

Algorithm Correctness

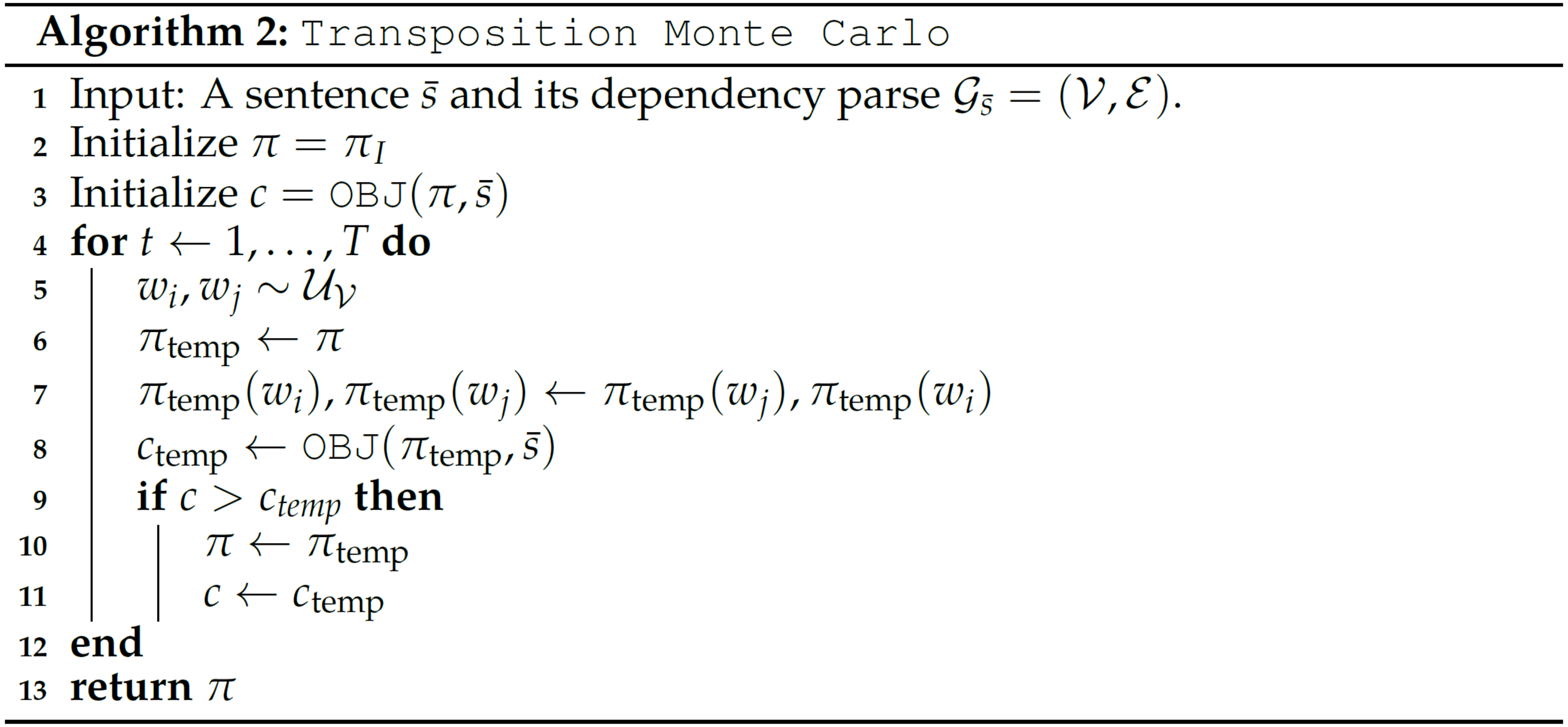
- Claim final child if odd number of children can always be placed with other odd children
- Final child needs to be placed on side that yields lower minLA score



Runtime analysis

- Runtime is claimed to be linear; their algorithm is worst-case $n \log n$
- Can be rectified using bucket-sort and data structures of Yannakakis, 1985

Additional Heuristics



```
Algorithm 2: Transposition Monte Carlo
Input: A sentence $\bar{s}$ and its dependency parse $\mathcal{G}_{\bar{s}} = (\mathcal{V}, \mathcal{E})$.
Initialize $\pi = \pi_I$
Initialize $c = \texttt{OBJ}(\pi, \bar{s})$
for $t \leftarrow 1, \ldots, T$ do
    $w_i, w_j \sim \mathcal{U}_{\mathcal{V}}$
    $\pi_{\text{temp}} \leftarrow \pi$
    $\pi_{\text{temp}}(w_i), \pi_{\text{temp}}(w_j) \leftarrow \pi_{\text{temp}}(w_j), \pi_{\text{temp}}(w_i)$
    $c_{\text{temp}} \leftarrow \texttt{OBJ}(\pi_{\text{temp}}, \bar{s})$
    if $c > c_{temp}$ then
        $\pi \leftarrow \pi_{\text{temp}}$
        $c \leftarrow c_{\text{temp}}$
end
return $\pi$
```

T is a parameter for balancing between retaining the standard word order and optimizing the objective function.

Small T – Similar to natural language, Large T – Locally optimal for objective

INTEGRATING NOVEL ORDERS WITH NLP

PART I: METHODS

- Order-Agnostic Methods
  - Word embeddings, topic models, bag-of-words text classifiers
  - Sentence encoders: DANs (Iyyer et al., 2015), SIF (Arora et al., 2017)
- Sequential Models
  - HMMs, MEMMs, CRFs
  - RNNs, LSTMs, GRUs, SRUs
- Position-Aware Methods
  - Transformers (Vaswani et al., 2017)
  - Encode ($w_1 \ldots w_n$) as $\{(w_1, 1), \ldots, (w_n, n)\}$ (Vinyals et al., 2016)

- Alternative word orders
  - Bidirectional models - ($w_1 \dots w_n$) and ($w_n \dots w_1$)
  - Permutation models – XLNet (Yang et al., 2019)

- Alignment
  - Attention (Bahdanau et al., 2015; Luong et al., 2015)
  - Preorders – Reorder source language to mimic target language order
  - Postorders – Reordering monotone translation to mimic target language order

Integrating Novel Orders

- Train models on permuted sentences instead of standard sentences
  - Surprisingly effective in initial experiments; does not leverage pretraining
- Input permuted sentences into standard pretrained encoders
  - Unsurprisingly, does not work
- Pretrain on permuted sentences
  - Not feasible, unclear if pretraining objects (e.g. language modelling) even make sense

Adapting pretrain-and-finetune

1. Embed sentence s using pretrained encoder to yield $(\boldsymbol{x_1} \dots \boldsymbol{x_n})$
2. Encode $(\boldsymbol{x_1} \dots \boldsymbol{x_n})$ using task-specific encoder to yield $(\boldsymbol{h_1} \dots \boldsymbol{h_n})$
3. Pool $(\boldsymbol{h_1} \dots \boldsymbol{h_n})$ to yield $\boldsymbol{h}$
4. Use $\boldsymbol{h}$ to make a prediction $\hat{y}$

pretrain-permute-finetune

1. Embed sentence s using pretrained encoder to yield $(\boldsymbol{x_1} \ldots \boldsymbol{x_n})$
2. Permute $(\boldsymbol{x_1} \ldots \boldsymbol{x_n})$ to yield $(\boldsymbol{z_1} \ldots \boldsymbol{z_n})$

   Permute
3. Encode $(\boldsymbol{z_1} \ldots \boldsymbol{z_n})$ using task-specific encoder to yield $(\boldsymbol{h_1} \ldots \boldsymbol{h_n})$
4. Pool $(\boldsymbol{h_1} \ldots \boldsymbol{h_n})$ to yield $\boldsymbol{h}$
5. Use $\boldsymbol{h}$ to make a prediction $\hat{y}$

INTEGRATING NOVEL ORDERS WITH NLP

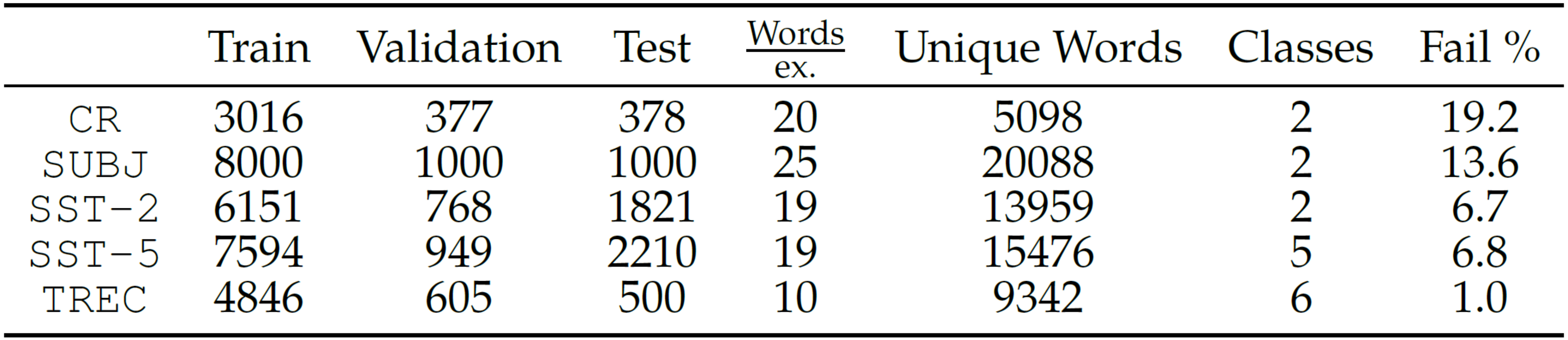
PART II: EXPERIMENTAL SETUP

Model

- spaCy dependency parser, frozen ELMo pretrained representations
- Bidirectional LSTM task-specific encoder with max-pooling and linear classifer
- Cross-Entropy Loss, Adam Optimizer, Dropout
- Train for 12 epochs (very similar to various early-stopping conditions)
- Mini-batch size of 16
- Aforementioned hyperparameters optimized for baselines
- Other hyperparameters (hidden size, dropout parameter) are independently optimized for every baseline/model we are considering.

- Orders
  - Random - r_r
  - Identity/standard English - r_I
  - Bandwidth (Cuthill-Mckee) - r_b
  - MinLA (corrected Gildea and Temperley) - r_m
  - Cutwidth (Yannakakis) - r_c
  - Bandwidth (Transposition Monte Carlo) - $r_{\tilde{b}}$
  - MinLA (Transposition Monte Carlo) - $r_{\tilde{m}}$
  - Cutwidth (Transposition Monte Carlo) - $r_{\tilde{c}}$

Datasets – English Language Text Classification



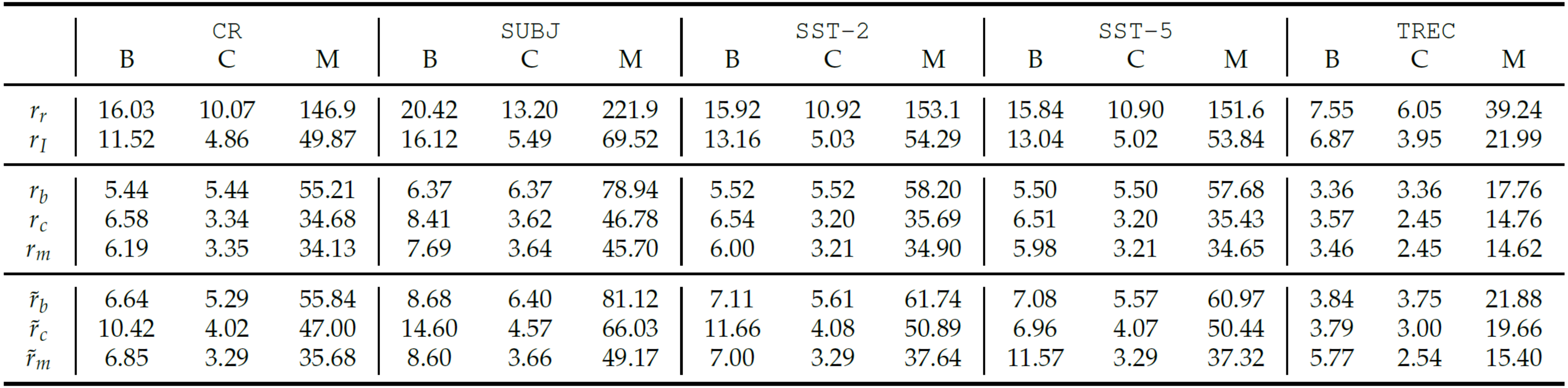
	Train	Validation	Test	$\frac{\text{Words}}{\text{ex.}}$	Unique Words	Classes	Fail %
CR	3016	377	378	20	5098	2	19.2
SUBJ	8000	1000	1000	25	20088	2	13.6
SST-2	6151	768	1821	19	13959	2	6.7
SST-5	7594	949	2210	19	15476	5	6.8
TREC	4846	605	500	10	9342	6	1.0

- **CR** : Sentiment analysis for customer reviews (Hu and Liu, 2004)
- **SUBJ** : Subjectivity analysis for movie data (Pang and Lee, 2004)
- **SST-2** : Sentiment analysis for movie reviews (Pang and Lee, 2005; Socher et al., 2013)
- **SST-5** : Fine-grained sentiment analysis for movie reviews (Pang and Lee, 2005; Socher et al., 2013)
- **TREC** : Question classification (Li and Roth, 2002)

INTEGRATING NOVEL ORDERS WITH NLP

PART III: RESULTS AND ANALYSIS

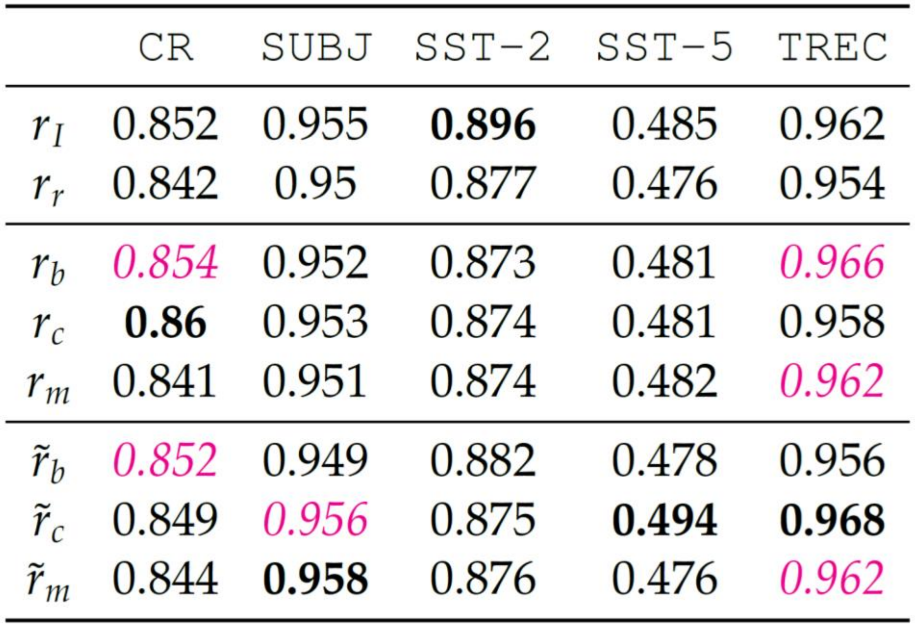
Optimization Results



	CR			SUBJ			SST-2			SST-5			TREC		
	B	C	M	B	C	M	B	C	M	B	C	M	B	C	M
r_r	16.03	10.07	146.9	20.42	13.20	221.9	15.92	10.92	153.1	15.84	10.90	151.6	7.55	6.05	39.24
r_I	11.52	4.86	49.87	16.12	5.49	69.52	13.16	5.03	54.29	13.04	5.02	53.84	6.87	3.95	21.99
r_b	5.44	5.44	55.21	6.37	6.37	78.94	5.52	5.52	58.20	5.50	5.50	57.68	3.36	3.36	17.76
r_c	6.58	3.34	34.68	8.41	3.62	46.78	6.54	3.20	35.69	6.51	3.20	35.43	3.57	2.45	14.76
r_m	6.19	3.35	34.13	7.69	3.64	45.70	6.00	3.21	34.90	5.98	3.21	34.65	3.46	2.45	14.62
$\tilde{r}_b$	6.64	5.29	55.84	8.68	6.40	81.12	7.11	5.61	61.74	7.08	5.57	60.97	3.84	3.75	21.88
$\tilde{r}_c$	10.42	4.02	47.00	14.60	4.57	66.03	11.66	4.08	50.89	6.96	4.07	50.44	3.79	3.00	19.66
$\tilde{r}_m$	6.85	3.29	35.68	8.60	3.66	49.17	7.00	3.29	37.64	11.57	3.29	37.32	5.77	2.54	15.40

- English substantially more optimal than random for (dataset, objective) pairs
- Substantial margin for all objectives to improve over English
- Algorithms from literature outperform our algorithms
- Gildea and Temperley, Yannakakis achieve similar results for all objectives
- Greedy optimization in Transposition Monte Carlo seems to work best for MinLA

- Downstream Results
  - English outperforms random
    - Margin is small
  - Orders due algorithms in literature:
    - Do not consistently outperform random
  - Orders due Transposition Monte Carlo:
    - Outperform random consistently (one exception is $r_{\tilde{b}}$ on **SUBJ** by 0.001)
    - Generally outperform corresponding order from algorithm in prior work
    - Outperform English for 4/5 datasets



	CR	SUBJ	SST-2	SST-5	TREC
r_I	0.852	0.955	**0.896**	0.485	0.962
r_r	0.842	0.95	0.877	0.476	0.954
r_b	*0.854*	0.952	0.873	0.481	*0.966*
r_c	**0.86**	0.953	0.874	0.481	0.958
r_m	0.841	0.951	0.874	0.482	*0.962*
$\tilde{r}_b$	*0.852*	0.949	0.882	0.478	0.956
$\tilde{r}_c$	0.849	*0.956*	0.875	**0.494**	**0.968**
$\tilde{r}_m$	0.844	**0.958**	0.876	0.476	*0.962*

- **TL;DR –** Reasonably convincing evidence that alternative orders may yield improvements over English

OUTCOMES

PART I: CONTRIBUTIONS

- Unifying treatments of linear order
  - Algorithmic reframing of prior psycholinguistic work
  - Extending set of orders considered in NLP in sequential models
- Considering alternative orders in NLP
  - Introduced pretrain-permute-finetune framework
  - Empirical evaluation of effects on downstream NLP; resolute evidence to warrant further inquiry into alternative orders

Goals:

- ☑Unified theory for linear word order
- ☑Improved downstream NLP systems

OUTCOMES

PART II: OPEN PROBLEMS AND FUTURE DIRECTIONS

- Beyond Dependencies
  - Information-theoretic approaches inspired by information locality
  - Constituency-based orderings / alternative syntactic theories
- Learned Orderings
  - End-to-end (differentiable) learning of orderings
  - Task-specific weighting (and corresponding combinatorial optimization algorithms)
- Understanding
  - Information obfuscation due to permutation
  - Interplay between pretrain and permute steps
  - Empirical understanding of when and why this works for downstream NLP

OUTCOMES

PART III: LIMITATIONS

- Dependence on Dependencies
  - Parsers in arbitrary languages cannot be assumed
  - Annotations/schemas do not exist for some languages (e.g. beyond UD)
- Rigid Optimization
  - No natural way to incorporate side-information
  - Meaningfulness of “pure” optimization is contingent on quality of parser
- Evaluation Settings
  - Tasks beyond sentence-level text classification; models beyond LSTMs
  - Languages beyond English

THANKS!

- Claire Cardie
- Bobby Kleinberg
- Tianze Shi [Cornell], Marty van Schijndel [Cornell], Forrest Davis [Cornell], Lillian Lee [Cornell], John Hewitt [Stanford], Percy Liang [Stanford], Dan Klein [Berkeley], Jason Eisner [JHU], Ge Gao [Cornell], Arzoo Katiyar [Cornell], Vlad Niculae [Cornell], Kai Sun [Cornell], Tal Linzen [NYU],Tatsu Hashimoto [Stanford], Sasha Rush [Cornell], Yoav Artzi [Cornell], Nelson Liu [Stanford], Nori Kojima [Cornell], Nick Tomlin [Berkeley]
- ACL SRW 2019 and NeurIPS 2019 Context and Compositionality reviewers and poster attendees
- Cornell NLP Fall 2019 Retreat audience
- Cornell CS and Cornell NLP