Back to Patterns:

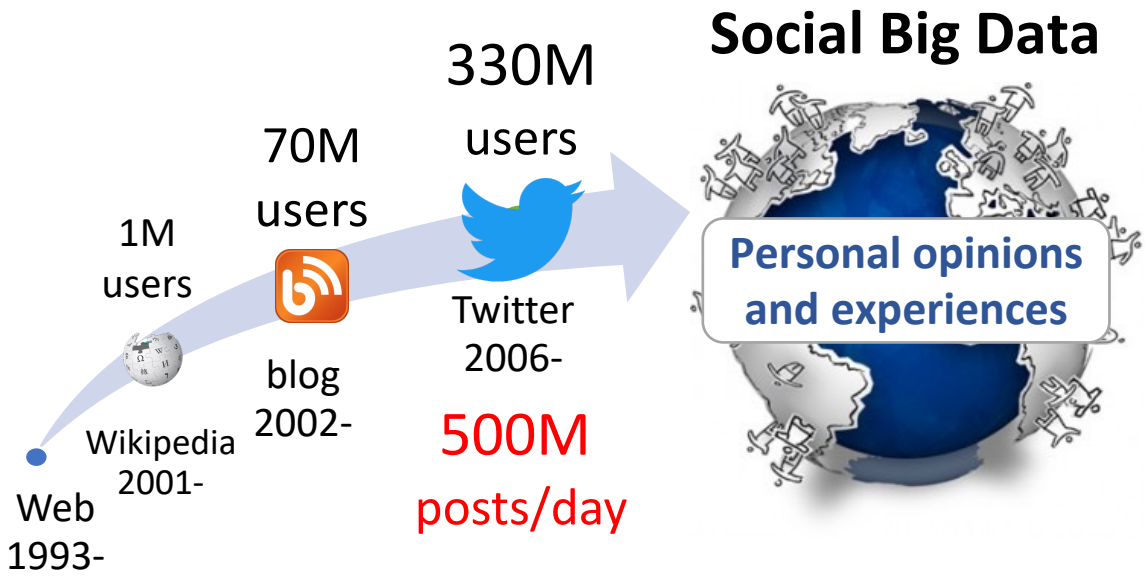
Efficient  Morphological Analysis with Feature-Sequence Trie

Naoki Yoshinaga

Institute of Industrial Science, The University of Tokyo

Whereas data is increasing, models become slower

- **Text data has been increasing** since the birth of the Web
  - SNS posts via smartphones
  - Communication via zoom/slack



- NLP **models become slower**, focusing on accuracy
  - *Efficient* neural methods are only *relatively efficient* and are not fast

Implementation of Japanese Morphological Analyzer (MA)	Speed [sents/s]	
Juman [Kurohashi+ 1994]	8802	non-neural
MeCab [Kudo+ 2004]	**52410**	non-neural
KyTea [Neubig+ 2011]	4892	non-neural
Juman++V1 [Morita+ 2015]	16	neural
Juman++V2 [Tolmachev+ 2018]	4803	neural

[Tolmachev+ 2018]

The outdated yet *sota efficient* methods have been used for ages to process the increasing textual data for sociolinguistics and marketing

Proposal: Pattern-based method for Japanese MA

- **Approach**: making pattern-based methods more accurate, instead of making neural methods more efficient

- **Proposal:** Pattern-based Japanese morphological analysis (MA)
  word segmentation, POS tagging, lemmatization
  - Regard segmentation and tagging as multi-class classifciation problem

  - Greedily solve this classification problem from left to right using patterns extracted from the training data and a dictionary

Avoid expensive argmax operations used in learning-based methods

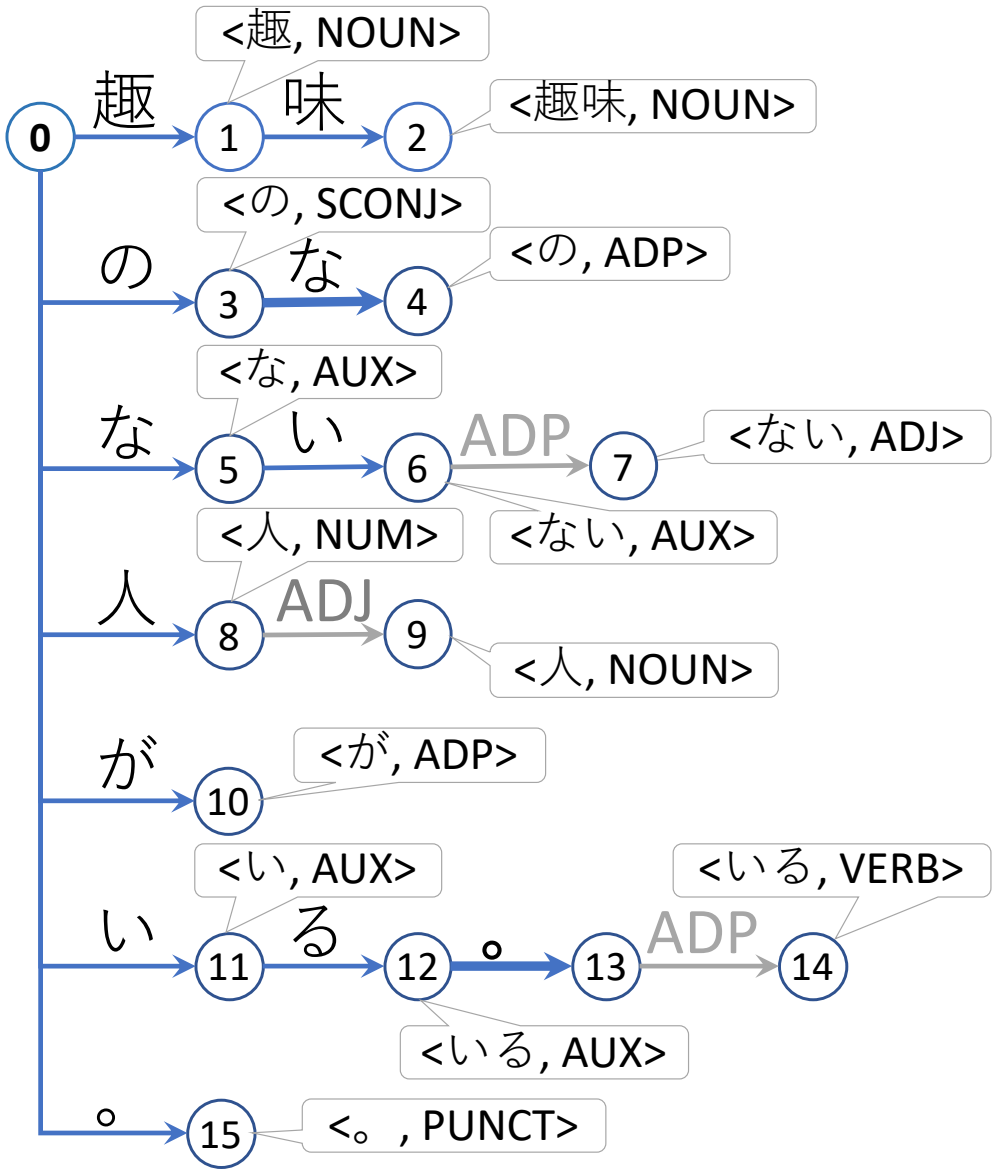
Running example (stop videos and look)

趣味のない人がいる。

shumi no nai hito ga iru .

Pattern	Word	POS (level 1)
趣味|	趣味	NOUN
の|な	の	ADP
ない|_ADP	ない	ADJ
人|_ADJ	人	NOUN
が|	が	ADP
いる|。_ADP	いる	VERB
。|	。	PUNCT

Feature-sequence trie (excerpted)



Results (excerpt)

- Compare our method (**Jagger**) to sota *efficient* learning-based methods (MeCab, Vibrato, Vaporetto) using the same dictionary
  - Environments: M2 MacBook Air with a 3.5-GHz CPU and 24-GB RAM

Method	Kyoto-U. Text Corpus (news)				Kyoto-U. Web Doc. Leads Corpus			
	speed [sent/s]	mem [MiB]	seg (F_1)	POS (F_1)	speed [sent/s]	mem [MiB]	seg (F_1)	POS (F_1)
MeCab	66,455	55.81	98.68	95.97	92,110	53.88	97.13	94.30
Vibrato	142,983	97.75	-	-	190,703	97.92	-	-
Vaporetto	117,767	658.80	98.94	96.92	200,823	642.63	97.35	94.08
Jagger	**1,007,344**	**26.39**	**98.73**	**96.55**	**1,524,305**	**28.89**	**97.17**	**94.20**

Jagger processes 1M sents/s with accuracy comparable to baselines

Takeaways

- Since **accuracies are becoming saturated** on NLP benchmarks, let's **focus more on underrepresented metrics, *e.g.*, efficiency**

- **Back to Patterns: Patterns are more powerful than you think**
  - It can rival learning-based methods in Japanese MA in terms of accuracy, and is 7-16x faster with 1/2-1/20 memory footprint

- **Take a speed-intensive approach** to **absolute efficiency in NLP**
  - Making very slow neural methods (slightly) fast seems uncompelling
  - Making a fast pattern-based method more accurate is compelling

Code: http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/jagger/