< IJCNLP 2008, Jan. 10th, 2008 >

Computing Paraphrasability of Syntactic Variants Using Web Snippets

Atsushi FUJITA and Satoshi SATO

Nagoya Univ., Japan

Automatic Paraphrasing

- Fundamental in NLP
  - Recognition: IR, IE, QA, Multi-Doc.Summarization
  - Generation: MT, TTS, Authoring aids
- Resources required
  - Handcrafted knowledge
    - Thesauri [Many work]
    - Transformation rules [Mel'cuk+, 87] [Dras, 99] [Jacquemin, 99]
  - Automatic knowledge acquisition
    - Distributional similarity [Lin+, 01] [Szpektor+, 04]
    - Aligning comparable/bilingual corpora [Many work]

Paraphrase Knowledge

- Template-like representation
  - Lexical paraphrases

    X wrote Y → X is the author of Y

    X solves Y → X deals with Y
  - Morpho-syntactic paraphrases (syntactic variants)

    X V Y → Y be v(Z)-PP by X Passivization

    X show a A Y → X v(Y) adv(A) Removing light-verb

- Lack of applicability conditions ⇒ incorrect results

Task Description

- Computing *paraphrasability* between phrases
  - Input: automatically generated paraphrase candidates
    - Pair of original and generated phrases (s and t)

We hurry checking it → We check it in a hurry

s t

  - Output: paraphrasability score [0,1]
    - Is t grammatical ?
    - Does t hold if s holds ? (semantic equivalence or inclusion)
    - Is t syntactically substitutable for s in some context ?

Issues and Solutions

- How to measure similarity between phrases ?
  - Contextual similarity: distributional similarity
    - Bag of words / Bag of dependency relations
  - Constituent similarity: handling syntactic variants
    - Syntactic transformation + Lexical derivation

- How to deal with data sparseness problem ?
  - Collect example sentences of phrases from Web snippets
    - Assessing grammaticality

Outline

1. Task Description
2. Paraphrases Handled
3. Proposed Method
4. Experiments
5. Discussion
6. Conclusion

Paraphrases of Predicate Phrases

- Symmetric vs. Asymmetric
  - X change Y ↔ X modify Y
  - X sprint → X run
- Equivalent / Inclusion / Entailment vs. Inference
  - X change Y ↔ X modify Y
  - X sprint → X run
  - X snore Y → X sleep Y
  - X married Y → X dated Y
- Lexical vs. Morpho-syntactic
  - X change Y ↔ X modify Y
  - X show a A Y → X v(Y) adv(A)

Paraphrases Handled

- Morpho-syntactic paraphrases (syntactic variants)
  - Syntactic transformation + Lexical derivation
    - Constituent similarity is guaranteed a little
    - e.g. Head-switching, Light-verb construction, Category-shift

Climate is in our favor → Climate is favorable for us

We hurry checking it → We check it in a hurry

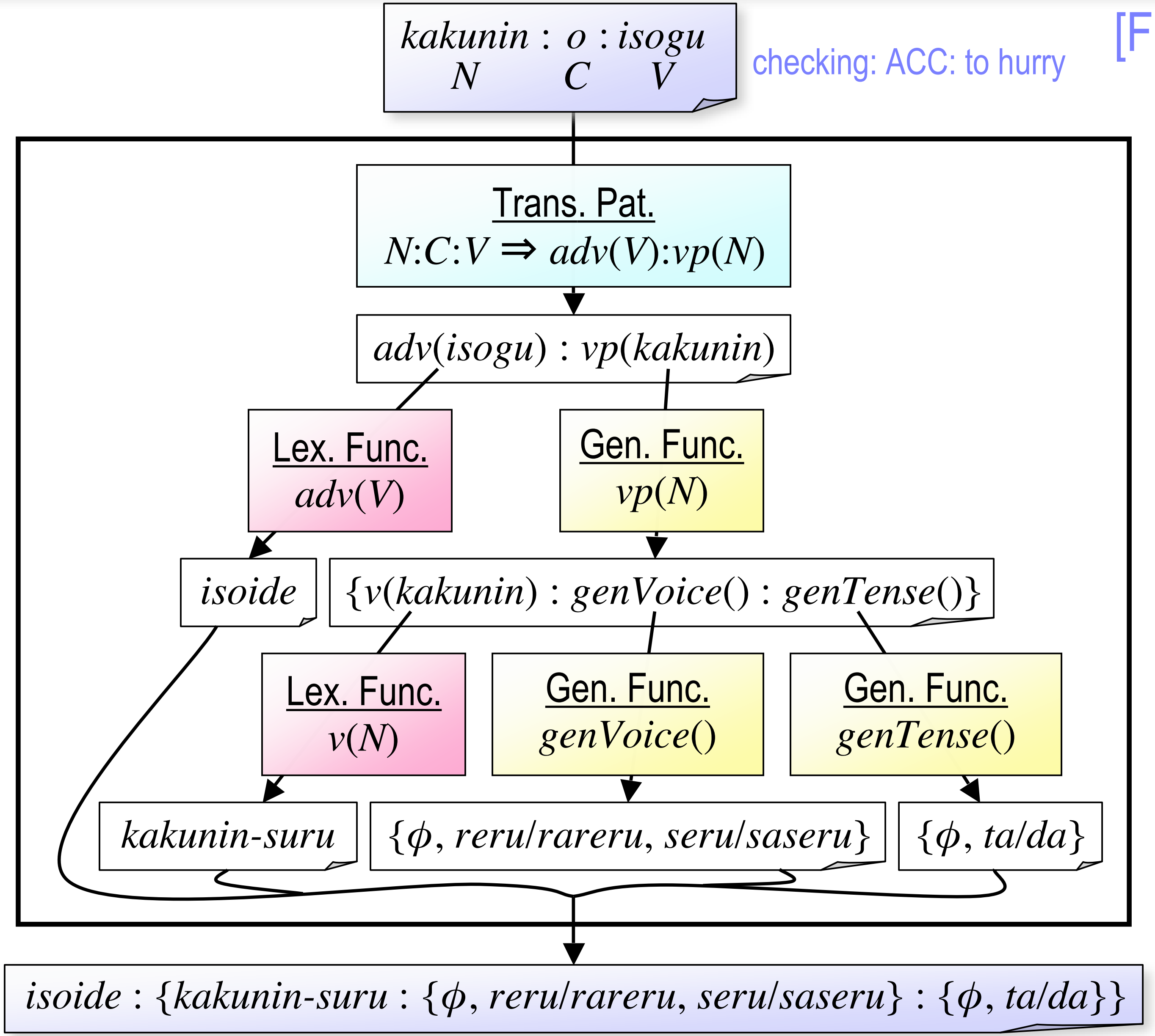
- Predicate phrases of Japanese

kakunin-o *isogu* → *isoide* *kakunin-suru*

checking-ACC	to hurry	in a hurry	to check
to hurry checking it		to check it in a hurry	

Syntactic Variant Generator for Japanese

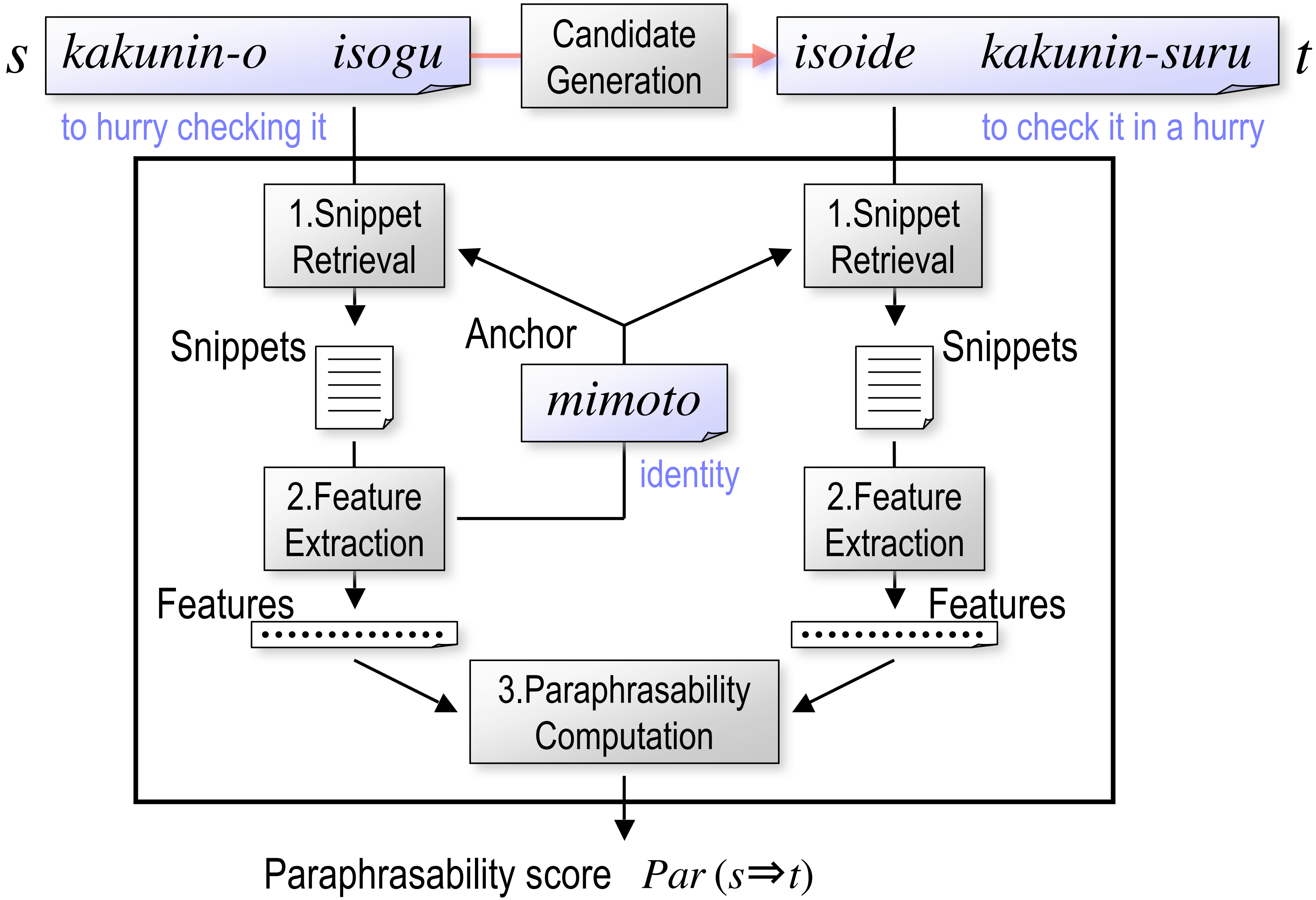
[Fujita+, 07]



Outline

1. Task Description
2. Paraphrases Handled
3. Proposed Method
4. Experiments
5. Discussion
6. Conclusion

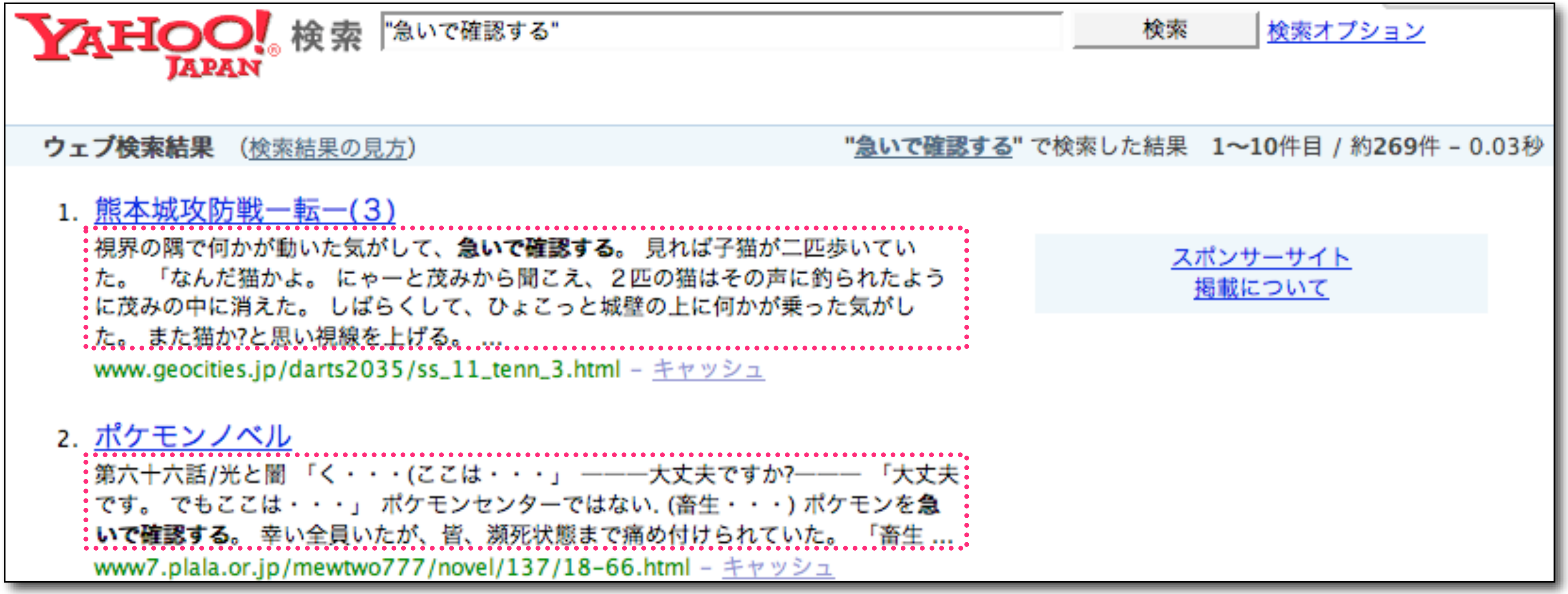
Overview



s
kakunin-o isogu
to hurry checking it
Candidate Generation
isoide kakunin-suru
t
to check it in a hurry
1.Snippet Retrieval
1.Snippet Retrieval
Anchor
Snippets
Snippets
mimoto
identity
2.Feature Extraction
2.Feature Extraction
Features
Features
3.Paraphrasability Computation
Paraphrasability score $Par(s \Rightarrow t)$

Step 1. Snippet Retrieval

- ``Phrase search''
  - Yahoo! JAPAN Web-search API
  - 500 top snippets



Step 2. Feature Extraction

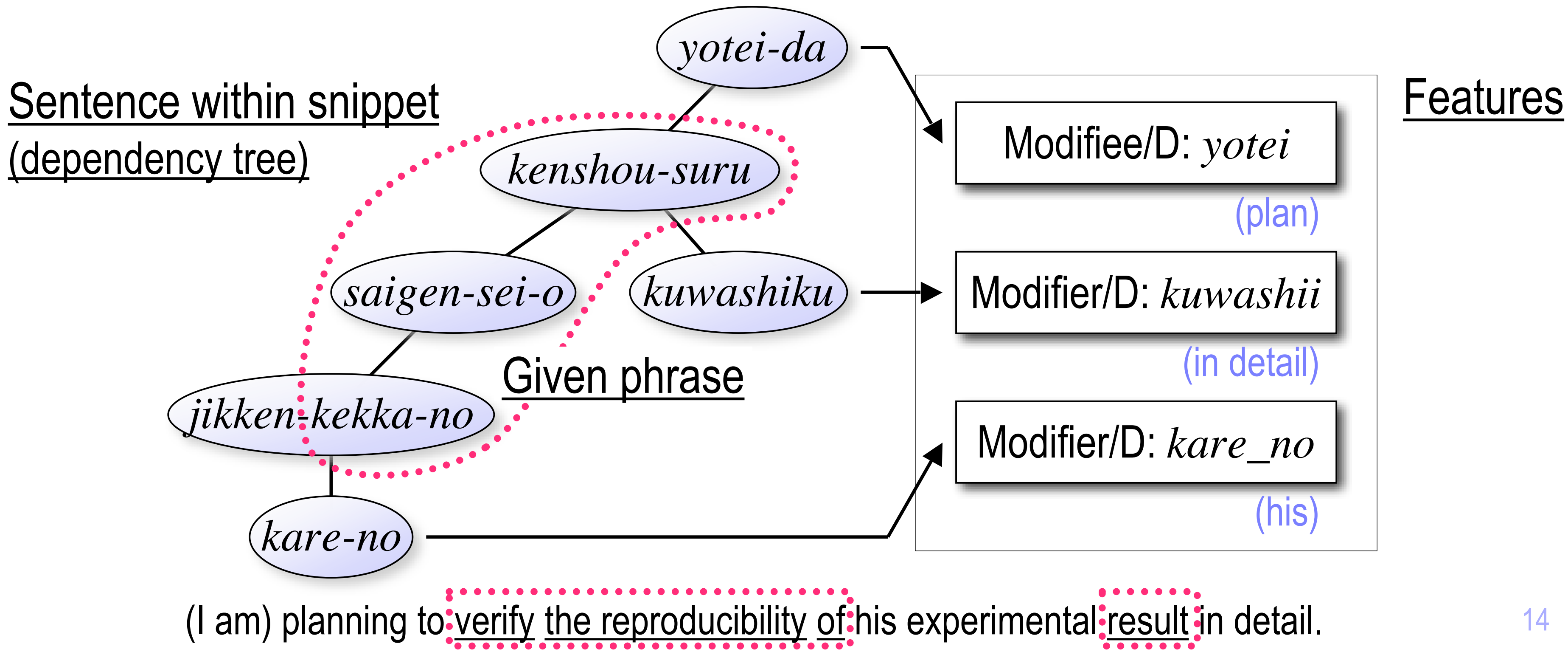
- HITS: # of pages Yahoo! API returns
  - Larger HITS ⇒ t is more likely grammatical

- BOW: content words around the phrase in snippets
  - BOWs surrounding s and t have similar distribution
    ⇒ s and t are semantically similar

- MOD: modifiers and modifiees of the phrase in snippets
  - s and t share a number of modifiers and modifiees
    ⇒ s and t are syntactically substitutable

Extracting MOD Features

- Modifier / modifiee chunk (*bunsetsu*)
  - Relation types (Depend / Appositive / Parallel)
  - Base form of the head-word (content word)
  - Some types of functional words (if any)



Step 2. Feature Extraction (Anc)

- Source-focused feature extraction
  - 1. Determine anchor a which strongly associates with s
    - Noun which most frequently modifies s (one of MOD features)

emi-o *ukaberu* (s) → *manmen* (a)

smile-ACC to show → from ear to ear

be smiling

doriburu-de *kake* *agaru* → *saido*

dribble-with to run to go up → side

to overlap by dribbling

  - 2. Retrieve snippets for s AND a and t AND a
  - 3. Extract BOW and MOD features from those snippets

Step 3. Paraphrasability Computation

- Lin: Lin's measure [Lin+, 01]

$$Par_{Lin}(s \Rightarrow t) = \frac{\sum_{f \in F_s \cap F_t} (w(s,f) + w(t,f))}{\sum_{f \in F_s} w(s,f) + \sum_{f \in F_t} w(t,f)}$$

  - F_s, F_t : Feature sets for s and t
  - $w(x, f)$: Weight of feature f in F_x (frequency in snippets)

- skew: α-skew divergence [Lee, 99]

$$\begin{aligned} Par_{skew}(s \Rightarrow t) &= \exp\left(-d_{skew}(t,s)\right) \\ d_{skew}(t,s) &= D\left(P_s \| \alpha P_t + (1-\alpha) P_s\right) \end{aligned}$$

  - $P_s = P(f \mid s)$, $P_t = P(f \mid t)$
  - α : approximation degree of KL divergence [0,1]

Summary

- Features: Contextual features of entire phrase
  - c.f. Marginal features [Torisawa, 06] [Pantel+, 07]
  - BOW, MOD
- Weight of features: Frequency in snippets
  - c.f. pair-wise MI [Lin+, 01] [Pantel+, 07]
  - c.f. Relative Focus Feature [Geffet+, 05]
- DS measures
  - Lin's measure (symmetric) [Lin+, 01]
  - α-skew divergence (asymmetric) [Lee, 99]

Outline

1. Task Description
2. Paraphrases Handled
3. Proposed Method
4. **Experiments**
5. Discussion
6. Conclusion

Setup: Candidate Generation

- 6 basic phrase types
- Most frequent 1,000+ phrases for each type
  - Mainichi newspaper corpus (1991-2005, 1.5GB)
  - Referring to dependency trees
- Syntactic variant generator for Japanese [Fujita+, 07]

Trans. Pat.	Gen. Func.	Lex. Func.
$N{:}C{:}V \Rightarrow adv(V){:}vp(N)$	$vp(N)$	$adv(V)$

Phrase type	# of tokens	# of types	th	types	Cov.(%)	Output		Ave.
$N : C : V$	20,200,041	4,323,756	1,000	1,014	10.7	1,536	(489)	3.1
$N_1 : N_2 : C : V$	3,796,351	2,013,682	107	1,005	6.3	88,040	(966)	91.1
$N : C : V_1 : V_2$	325,964	213,923	15	1,022	12.9	75,344	(982)	76.7
$N : C : Adv : V$	1,209,265	923,475	21	1,097	3.9	8,281	(523)	15.7
$Adj : N : C : V$	378,617	233,952	20	1,049	14.1	128	(50)	2.6
$N : C : Adj$	788,038	203,845	86	1,003	31.4	3,212	(992)	3.2
Total	26,698,276	7,912,633		6,190		176,541	(4,002)	44.1

Examples of Phrases

N:C:V

kakunin-o isogu

checking-ACC to hurry

to hurry checking it

N:C:Adv:V

kentou-o sarani susumeru

consideration-ACC further to go ahead

to take consideration further

N_1:N_2:C:V

songai-baisho-o motomeru

damage-reparation-ACC to demand

to demand reparation for damage

N:C:Adj

nodo-ga itai

throat-NOM be painful

to have a sore throat

N:C:V_1:V_2

toukei-o tori-hajimeru

statistics-ACC to take-to start

to start collect statistics

Adj:N:C:V

takai hyouka-o ukeru

high assessment-ACC to receive

to be rated high

Setup: Computing Paraphrasability Scores

- 15 measures:
  - Proposed: {HITS,{BOW,MOD,HAR}×{Lin,skew}}×{Nor,Anc}
  - BL (Mainichi): HITS using 1.5GB newspaper corpus

	Nor.HITS			Nor.BOW.*			Nor.MOD.*		
Phrase type	Output		Ave.	Output		Ave.	Output		Ave.
$N : C : V$	1,405	(489)	2.9	1,402	(488)	2.9	1,396	(488)	2.9
$N_1 : N_2 : C : V$	9,544	(964)	9.9	9,249	(922)	10.0	8,652	(921)	9.4
$N : C : V_1 : V_2$	3,769	(876)	4.3	3,406	(774)	4.4	3,109	(762)	4.1
$N : C : Adv : V$	690	(359)	1.9	506	(247)	2.0	475	(233)	2.0
$Adj : N : C : V$	45	(20)	2.3	45	(20)	2.3	42	(17)	2.5
$N : C : Adj$	1,459	(885)	1.6	1,459	(885)	1.6	1,399	(864)	1.6
Total	16,912	(3,593)	4.7	16,067	(3,336)	4.8	15,073	(3,285)	4.6

	Anc.HITS			Anc.BOW.*			Anc.MOD.*			Mainichi		
Phrase type	Output		Ave.	Output		Ave.	Output		Ave.	Output		Ave.
$N : C : V$	1,368	(488)	2.8	1,366	(487)	2.8	1,360	(487)	2.8	1,103	(457)	2.4
$N_1 : N_2 : C : V$	7,437	(897)	8.3	7,424	(894)	8.3	6,795	(891)	7.6	3,041	(948)	3.2
$N : C : V_1 : V_2$	2,517	(697)	3.6	2,497	(690)	3.6	2,258	(679)	3.3	1,156	(548)	2.1
$N : C : Adv : V$	342	(174)	2.0	339	(173)	2.0	322	(168)	1.9	215	(167)	1.3
$Adj : N : C : V$	41	(18)	2.3	41	(18)	2.3	39	(16)	2.4	14	(7)	2.0
$N : C : Adj$	1,235	(809)	1.5	1,235	(809)	1.5	1,161	(779)	1.5	559	(459)	1.2
Total	12,940	(3,083)	4.2	12,902	(3,071)	4.2	11,935	(3,020)	4.0	6,088	(2,586)	2.4

Evaluation 1: Ev.Gen

- Question
  - Can a correct paraphrase have the highest score among candidates for a source phrase ?

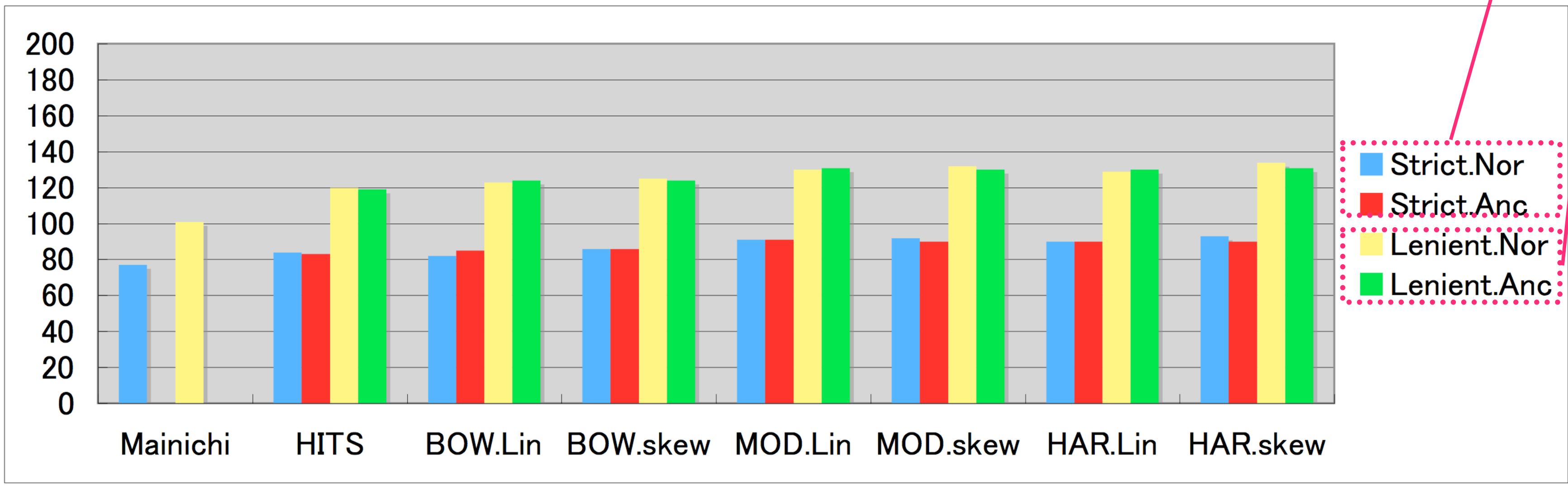
- Judgment (2 assesors)
  - For 200 input, the best candidates of 15 models

Results 1: Ev.Gen

- Mainichi **<<** *.HITS ≒ *.BOW.* < *.MOD.* ≒ *.HAR.*
  - Web enables us to compute paraphrasability accurately
- Candidates with higher scores are more likely correct
  - e.g. Lenient Prec. over 93% (th=0.5)
- Nor.X.* ≒ Anc.X.* (discuss later)

1 or 2 judges' OK

2 judges' OK



Evaluation 2: Ev.Rec

- Question
  - How is the method useful for collecting paraphrase instances ?

- Judgment (2 assesors)
  - 200 best candidates for each of 15 models

Results 2: Ev.Rec

- Mainichi ≒ *.HITS << *.BOW.* ≒ *.MOD.* ≒ *.HAR.*
  - DS measures outperformed HITS
  - Lenient Prec. almost reach a ceiling
- Nor.X.* ≒ Anc.X.* again
  - Anchor selection might be inappropriate
  - 2 or more content words make s rarely ambiguous

Results 2: Ev.Rec

- Remaining problems
  - Dropping N_1 from $N_1{:}N_2{:}C{:}V$
    - Typically functions as generalization

yagai-konsato → *konsato* **o**

outdoors-concert — concert

    - N_1 sometimes plays as the semantic head of $N_1{:}N_2$

shukketsu-taryou-de shibou-suru → *taryou-de shibou-suru* **x**

blood loss-plenty-ABL to die — loss-plenty-ABL to die

die due to heavy blood loss — (*) die due to plenty

  - Solutions:
    - Semantic parsing, Phrase boundary detection, etc.

Outline

Discussion: Issues Addressed

- Measurement of paraphrasability between phrases
  - Reasonably nice (Ev.Gen: over 65%, Ev.Rec: over 96%)
    - Combining constituent and contextual similarities
  - Room for improvement
    - Feature selection [Hagiwara+, 08]
    - Feature weighting [Lin+, 01] [Geffet+, 05]

- Data sparseness problem
  - Not perfectly solved
  - TSUBAKI offers larger number of snippets [Shinzato+, 08]

Discussion: Technical Issues

- Coverage
  - For 50% input, no candidate is output
  - More robust generation system
    - To generate a wider range of paraphrases
    - To handle other types of phrases with less human-labor
- Portability
  - 90% of candidates are filtered out due to 0 HITS
  - Use SLMs to prune incorrect candidates before querying

Conclusion

- Computing *paraphrasability* between phrases
  - Input: paraphrase candidates
    - Automatically generated
    - Syntactic variants
    - Predicate phrases in Japanese
  - Output: paraphrasability score [0,1]
    - Is t grammatical ?
    - Does t hold if s holds ? (semantic equivalence or inclusion)
    - Is t syntactically substitutable for s in some context ?
- Proposed method achieved reasonable results
  - Ev.Gen: over 65% (over 93% w/ th=0.5), Ev.Rec: over 96%