PARAPHRASES FOR TEXTUAL INFERENCE

Olga Kukina, University of Saarland

SS07: Linguistic Inference and Textual Entailment

OUTLINE

- Intro
  - what are paraphrases?
  - why to automatically learn paraphrases?
  - NLP applications
- Phrase-level paraphrasing
  - DIRT – algorithm (D. Lin & P.Pantel)
- Sentence-level paraphrasing
  - MSA approach (R.Barzilay & L. Lee)
- Paraphrasing with bilingual corpora
  - approach by C.Bannard & C.Callison-Burch
- Conclusion

WHAT ARE PARAPHRASES?

- Paraphrases are alternative ways to convey the same information
  - phrase paraphrases:

    Microsoft's monopoly – monopoly of Microsoft
  - sentence paraphrases:

Prof. Pinkal gives a course on Textual entailment and also delivers another one on Semantic Theory →
Prof.Pinkal is the lecturer for the courses on Textual Entailment and Semantic Theory.

- Paraphrases as inference rules:

  X caused Y → *Y is blamed on X*

WHY TO AUTOMATICALLY LEARN PARAPHRASES?

- traditional knowledge bases: manually created inference rules
  - extremely laborious
  - hard to account for all possible paraphrases:
  - *eg. X asks for Y*

MANUAL VARIATIONS	DIRT VARIATIONS
X wants *Y*; *X* requests *Y*; *X*'s request for *Y*	*X* requests *Y*; *X* seeks *Y*; *X*'s request for *Y*; *X* obtains *Y*; *Y* is requested by *X*; *X* solicits *Y*; *X* requests for *Y*; *X* demands *Y*; *X* pleads for *Y*; *X* wants *Y*; *X* presses for *Y*; *X* appeals for *Y*; ...

LEARNING PARAPHRASES: APPLICATIONS

- IR
- IE
- QA

recognizing mismatch between expressions in query and expressions in texts

  + eg. Who is the author of "Star Spangled Banner"?
    - ... Francis Scott Key wrote the .Star Spangled Banner in 1814.
    - ...comedian-actress Roseanne Barr sang ... Star Spangled Banner before a San Diego Padres-Cincinnati Reds game.

? same ranking

LEARNING PARAPHRASES: APPLICATIONS

- NL generation:
  - *creation of more varied and fluent text*
- Multidocument summarization:
  - *condensing information repeated across documents*
- Automatic evaluation of machine translation :
  - *identifying alternative and equally valid ways of translating a text*
- RTE:
  - *identify if the hypothesis is actually a paraphrase of (some part of) the text*

ALGORITHM 1

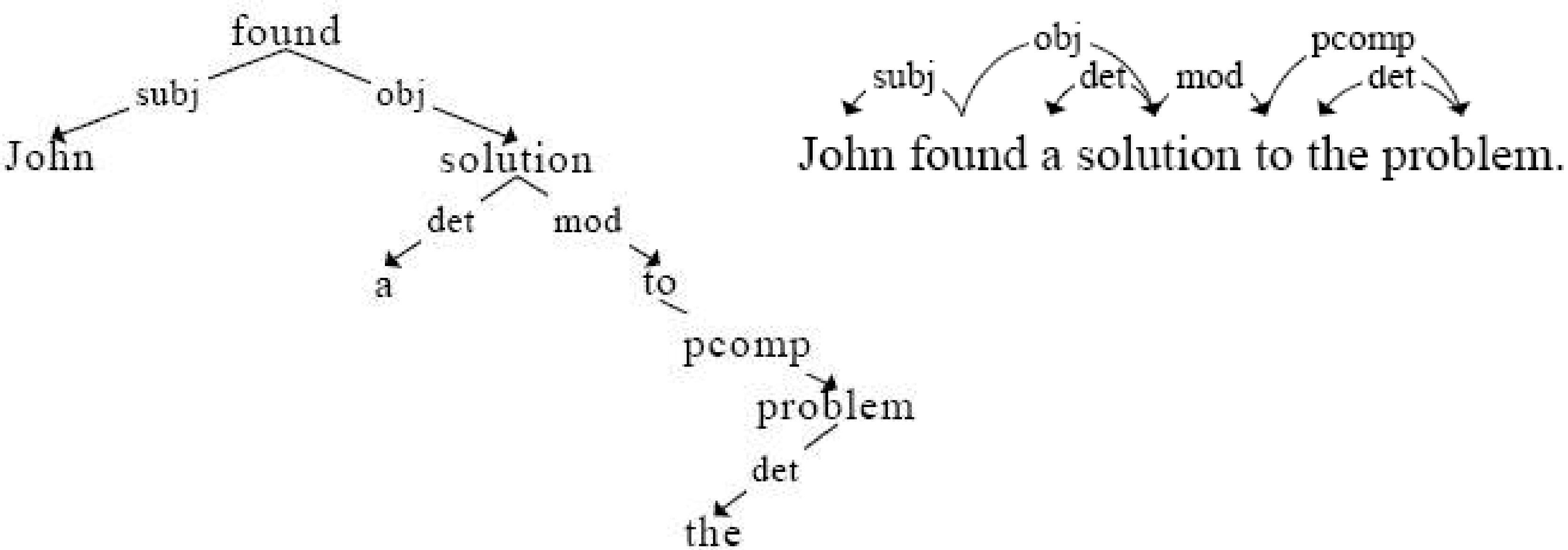
DISCOVERING INFERENCE RULES FROM TEXT

PHRASE-LEVEL PARAPHRASING : DIRT

- Unsupervised algorithm for Discovering Inference Rules from Text (DIRT)
  - Dekang Lin & Patric Pantel, Uni of Alberta, Canada
- Generalization of algorithm for finding similar words
  - Based on Distributional Hypothesis:
    - words that occurred in the same contexts tend to have similar meanings (Harris, 1985)
- Distributional Hypothesis applied to dependency trees
  - if two paths tend to link the same sets of words→their meanings are similar

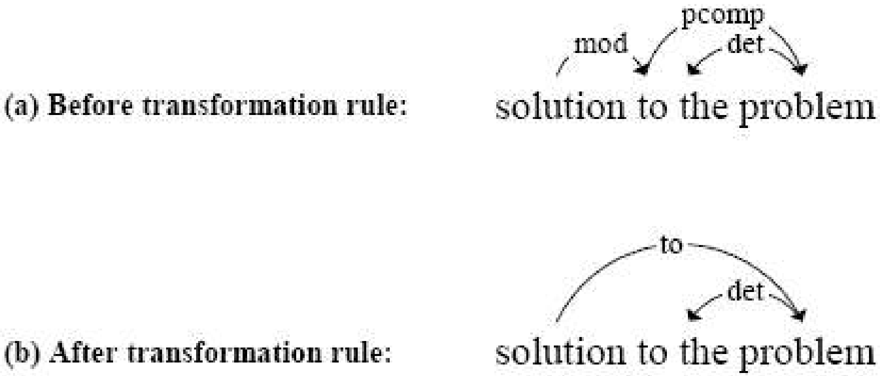
DIRT : PATH EXTRACTION

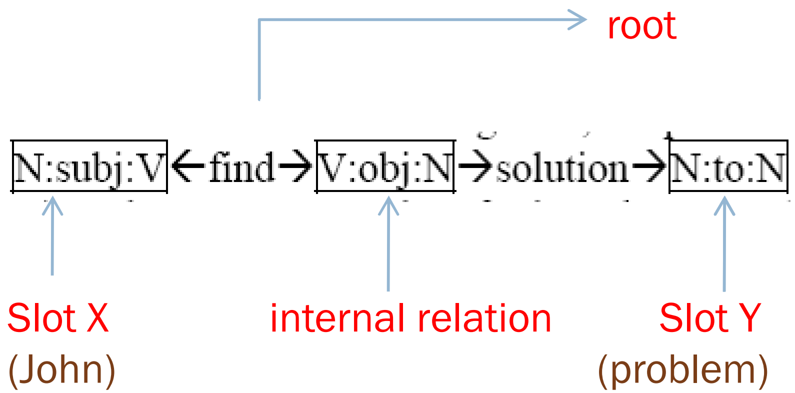
- Path : binary relation between 2 entities in dependency tree
- Minipar for generating dependency trees



DIRT : PATH EXTRACTION

- transformation rule
  - connect the prepositional complement directly to the words modified by the preposition
- Paths: representations of indirect semantic relations between two content words





DIRT : PATH EXTRACTION

- Constraints on the paths:

  N:subj:V←find→V:obj:N→solution→N:to:N

  - slot fillers must be nouns (slots=variables in inference)
  - any dependency relation that does not connect two content words is excluded from a path
  - the frequency count of an internal relation must exceed a threshold
  - an internal relation must be between a verb and an object-noun or a small clause

DIRT: SIMILARITY B/T PATHS

- Extended Distributional Hypothesis:
  - If two paths tend to occur in similar contexts, the meanings of the paths tend to be similar.

"X finds a solution to Y"		*"X solves Y"*	
SLOTX	SLOTY	SLOTX	SLOTY
commission	strike	committee	problem
committee	civil war	clout	crisis
committee	crisis	government	problem
government	crisis	he	mystery
government	problem	she	problem
he	problem	petition	woe
I	situation	researcher	mystery
legislator	budget deficit	resistance	crime
sheriff	dispute	sheriff	murder

DIRT : ALGORITHM STEPS

1. Collect path and slot filler frequencies from corpus
2. Create a triple database to store frequencies
3. Compute mutual information
4. Compute similarity b/t a pair of paths
5. Find the most similar paths

DIRT: TRIPLE DATABASE

- frequencies of:
  - all paths
  - slot fillers
    - (SlotX,w_1), (SlotY,w_2) – features of path p
- triple database

frequency

X pulls body from Y:

SlotX:			
	diver	1	2.45
	equipment	1	1.65
	police	2	2.24
	rescuer	3	4.84
SlotY:			
	bus	2	3.09
	coach	1	2.05
	debris	1	2.36
	feet	1	1.75

Mutual info b/t a slot & a filler

DIRT: MUTUAL INFORMATION

- measure for the association strength between <u>two</u> words

$$mi(x,y) = \log \frac{P(x,y)}{P(x)P(y)}$$

- handling <u>three</u> events

$$mi(x,y,z) = \log \frac{P(x,y,z)}{P(x)P(y)P(z)}$$

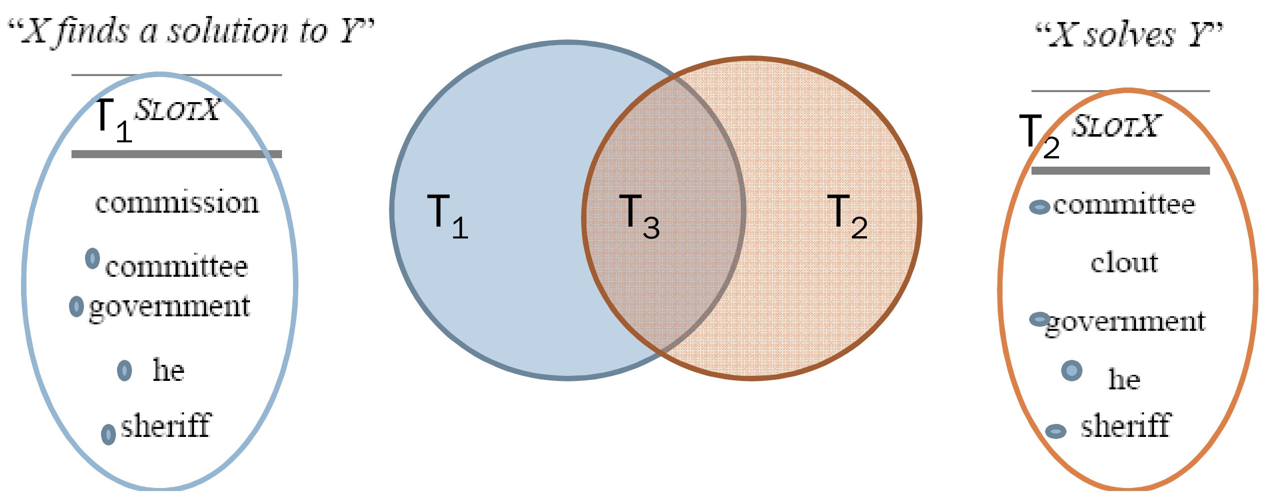
- more accurately

$$\boxed{mi(p,Slot,w) = log \frac{P(p,Slot,w)}{P(Slot)P(p \mid Slot)P(w \mid Slot)}}$$

DIRT: SIMILARITY B/T A PAIR OF SLOTS

- two paths have high similarity if there are a large number of common features
- similarity between a pair of slots : $slot_1=(p_1,s)$ and $slot_2=(p_2,s)$

$$sim(slot_1, slot_2) = \frac{\sum_{w \in T(p_1,s) \cap T(p_2,s)} mi(p_1,s,w) + mi(p_2,s,w)}{\sum_{w \in T(p_1,s)} mi(p_1,s,w) + \sum_{w \in T(p_2,s)} mi(p_2,s,w)}$$



DIRT: MUTUAL INFORMATION

$$mi(p, Slot, w) = log \frac{P(p, Slot, w)}{P(Slot)P(p \mid Slot)P(w \mid Slot)}$$

$$mi(p, Slot, w) = \log \frac{\frac{|p, Slot, w|}{|*,*,*|}}{\frac{|*, Slot, *|}{|*,*,*|}\frac{|p, Slot, *|}{|*, Slot, *|}\frac{|*, Slot, w|}{|*, Slot, *|}} = \log \frac{|p, Slot, w| \times |*, Slot, *|}{|p, Slot, *| \times |*, Slot, w|}$$

DIRT : COMPUTE SIMILARITY B/T PATHS

- similarity between a pair of paths p_1 *and* p_2 *is the geometric average* of the similarities of their *slots : SlotX and SlotY*

$$S(p_1, p_2) = \sqrt{sim(SlotX_1, SlotX_2) \times sim(SlotY_1, SlotY_2)}$$

DIRT: FINDING MOST SIMILAR PATHS

- TASK: find similar paths to "X solves Y"
- Retrieve candidate paths for *p*
- For each candidate path *c, count the number of features shared by c* and *p.*
- *Filter out c if the number of its common features with p is less* than 1% of the total number of features for *p* and *c.*
- Compute the similarity between *p and the candidates that passed the* filter
- Sort paths in descending order

DIRT: FINDING MOST SIMILAR PATHS

- most similar paths to *X solves Y*

1.	Y is solved by X	26.	X clears up Y
2.	X resolves Y	27.	*X creates Y
3.	X finds a solution to Y	28.	*Y leads to X
4.	X tries to solve Y	29.	Y is eased between X
5.	X deals with Y	30.	X gets down to Y
6.	Y is resolved by X	31.	X worsens Y
7.	X addresses Y	32.	X ends Y
8.	X seeks a solution to Y	33.	*X blames something for Y

DIRT: EVALUATION

- ran Minipar on 1Gb newspaper corpus
  - 2 million paths → triple database
- took first 15 questions from TREC 8 and extracted

  paths
- ran the DIRT algorithm *f*or each path to compute its Top-40 most similar paths
- classified each extracted path as *correct* or *incorrect* :
  - *eg p =“X manufactures Y” & p’ = “X’s Y factory”*
- compare correct DIRT output to a set of human-generated paraphrases

DIRT: EVALUATION

Question #	Question
Q_1	Who is the author of the book, "The Iron Lady: A Biography of Margaret Thatcher"?
Q_2	What was the monetary value of the Nobel Peace Prize in 1989?
Q_3	What does the Peugeot company manufacture?
Q_4	How much did Mercury spend on advertising in 1993?

Question	Paths	Manual	DIRT (correct)	Intersection	Accuracy
Q_1	X is author of Y	7	21	2	52.5%
Q_2	X is monetary value of Y	6	0	0	0%
Q_3	X manufactures Y	13	37	4	92.5%
Q_4	X spend Y	7	16	2	40.0%
	spend X on Y	8	15	3	37.5%

DIRT: OBSERVATIONS

- DIRT generally extracted more paraphrases than humans
- little overlap between the automatically and manually generated paraphrases
  - →finding useful inference rules is very difficult for humans as well as machines
- Better performance for paths with verb roots than for paths with noun roots
- No paths for 3 questions
- Possible improvements:
  - Accounting for polarity:
    - *cp "X worsens Y" & "X solves Y"*
  - Using semantic classes to extend paths with constraints on the variables.

ALGORITHM 2
MULTIPLE - SEQUENCE ALIGNMENT APPROACH

APPROACH USING MSA: MOTIVATION

- Paraphrasing larger units:
  - cannot rely only on domain-independent lexical resources
  - paraphrasing smaller lexical units is not enough

After the latest Fed rate cut, stocks rose across the board.
Winners strongly outpaced losers after Greenspan cut interest rates again.

- need for special sentence-level paraphrasing

APPROACH USING MSA : INTRO

- Multiple-sequence alignment (MSA) algorithm
  - R.Barzilay & L.Lee, Cornell University, USA

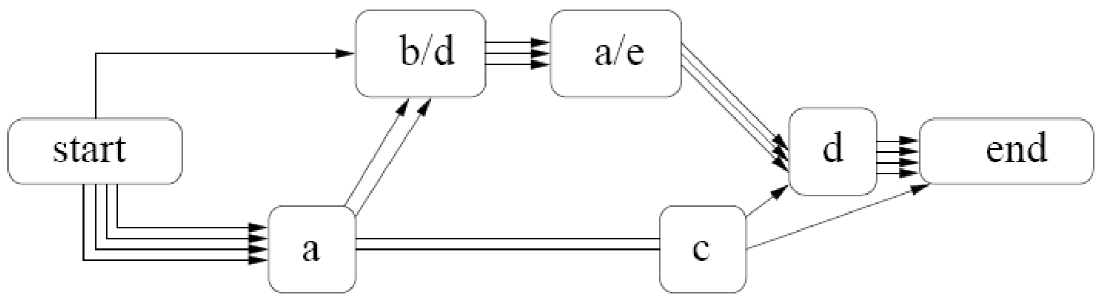
- Features:
  - Focus on paraphrase generation
  - Flexible paraphrase types
  - Use of comparable corpora
  - Minimal use of knowledge resources

MULTIPLE-SEQUENCE ALIGNMENT

- Input: n strings/sequences
- Output: n-row correspondence table
  - rows correspond to sequences
  - columns indicate the elements corresponding to that point
- A lattice may be generated from the MSA

a	b	a	_	d
a	_	_	c	d
a	_	_	c	_
_	b	a	_	d
a	d	e	_	d

MSA

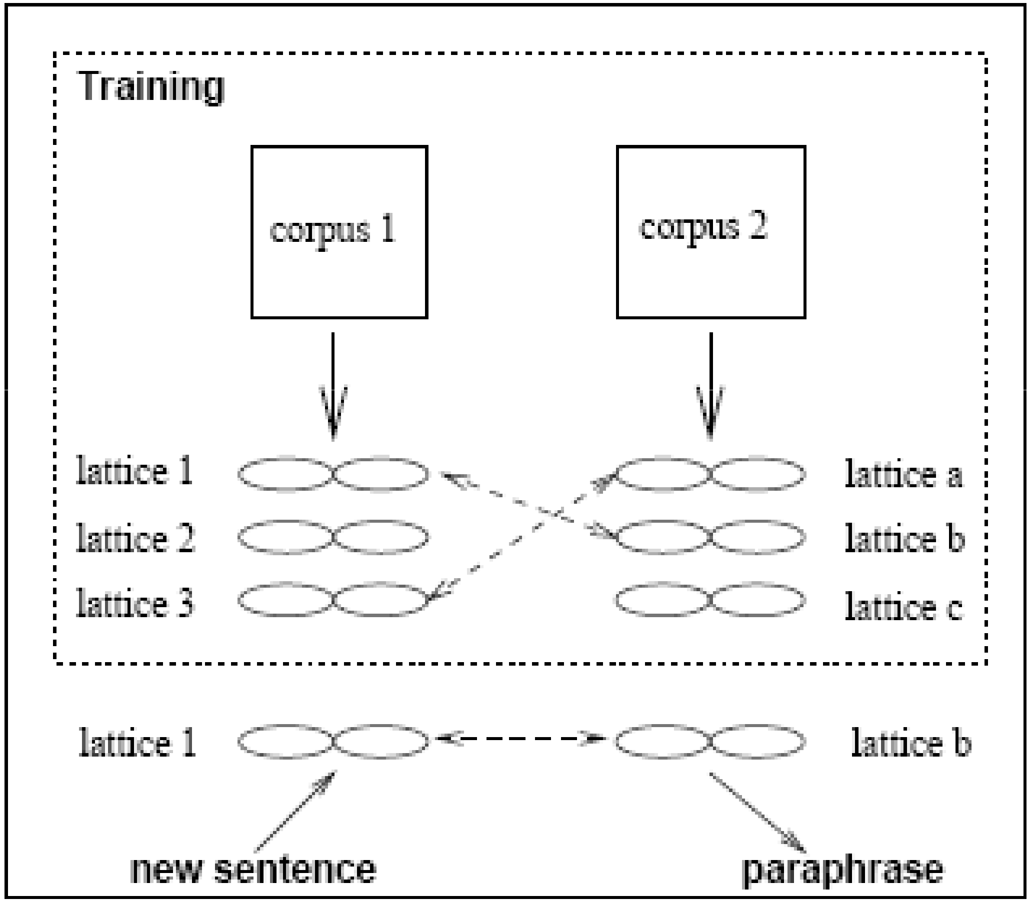


LATTICE

APPROACH USING MSA: ALGORITHM

- Start with two comparable corpora
- Identify recurring patterns in each dataset
  - *X (injured/wounded) Y people, Z seriously*
- Identify pairs of patterns across the two data sets that represent paraphrases
  - *Y were (wounded/hurt) by X,among them Z were in serious condition*

APPROACH USING MSA: ARCHITECTURE



APPROACH USING MSA: SENTENCE CLUSTERING

- clustering of sentences
  - Similarity metric: word n-gram overlap (n=1,2,3,4)
  - Proper nouns, dates and numbers replaced by generic tokens

A Palestinian suicide bomber blew himself up in a southern city Wednesday, killing two other people and wounding 27.

A suicide bomber blew himself up in the settlement of Efrat, on Sunday, killing himself and injuring seven people.

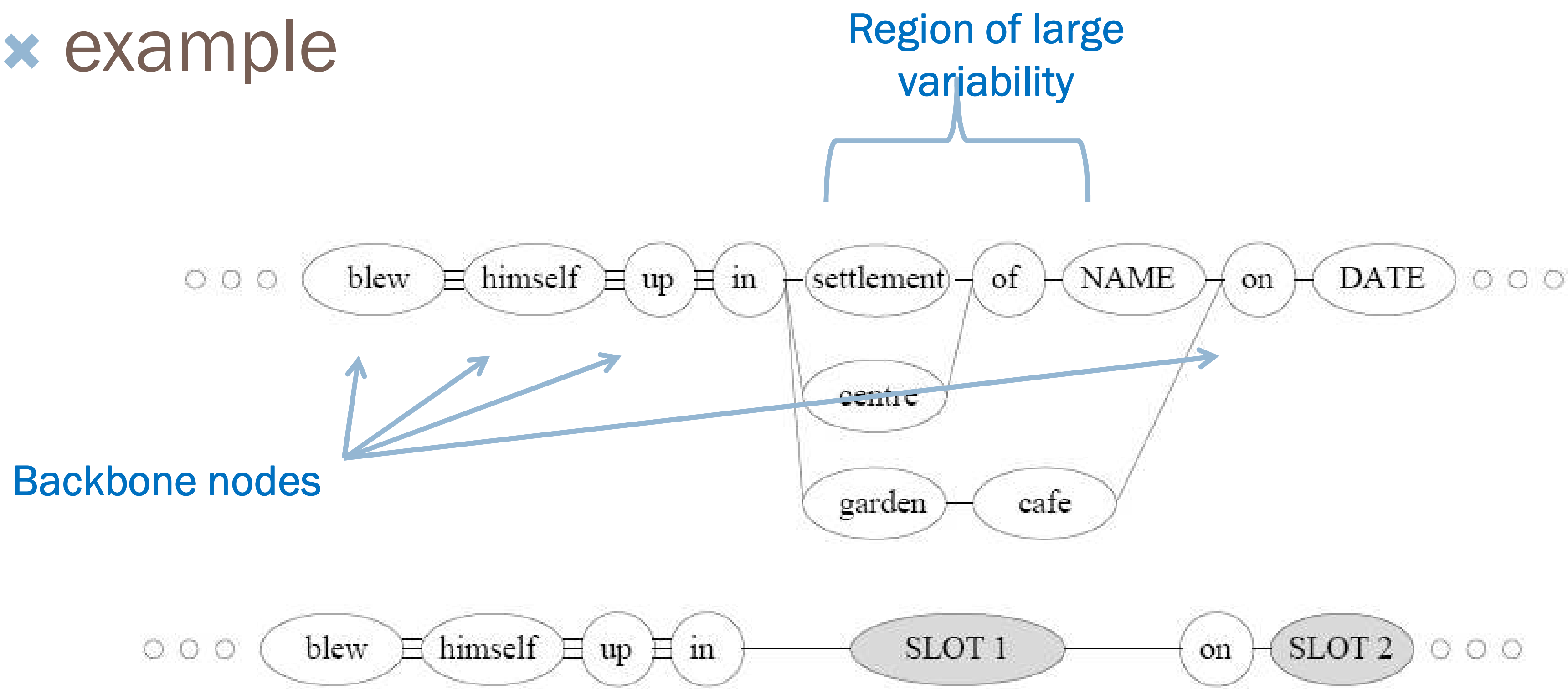
A suicide bomber blew himself up in the centre of Netanya on Sunday, killing three people as well as himself and injuring 40.

APPROACH USING MSA: LATTICES

- Compute *MSA* & Generate lattices:
  - Number of edges between nodes corresponds to number of sentences following that path
  - Identify Backbone Nodes
    - Nodes shared by more than 50% of the cluster's sentences
  - Identify regions of variability:
    - Argument variability: replace by slots
    - Synonym variability: to be preserved

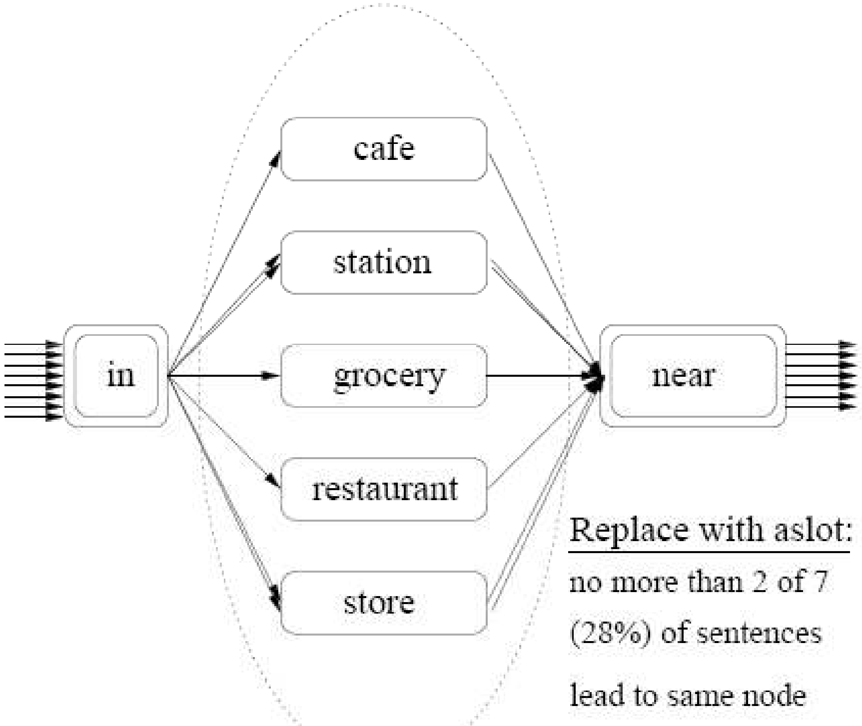
APPROACH USING MSA: LATTICES

- Compute MSA & generate lattices
- example



APPROACH USING MSA: VARIABILITY

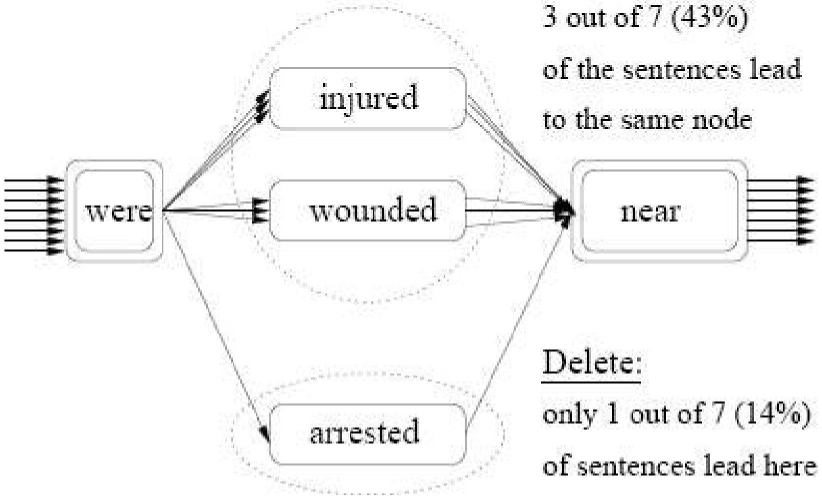
ARGUMENT VARIABILTY



Replace with aslot:
no more than 2 of 7
(28%) of sentences
lead to same node

SYNONYM VARIABILTY

Preserve both nodes:
3 out of 7 (43%)
of the sentences lead
to the same node



Delete:
only 1 out of 7 (14%)
of sentences lead here

APPROACH USING MSA: MATCHING LATTICES

- Parallel corpora
  - Sentence alignment
- Comparable corpora
  - Take pair of lattices from 2 corpora
  - Look back at clusters they came from
  - Compare slot values of sentences from articles on same date & topic
    - Paraphrases take same argument values

$slot_1$ bombed $slot_2$

the Israeli fighters bombed Gaza strip

$slot_3$ was bombed by $slot_4$

Gaza strip was bombed by the Israeli fighters

APPROACH USING MSA: GENERATING PARAPHRASES

- Input: sentence to be paraphrased, X
- Check if exists lattice XX to represent X
- If XX exists, retrieve lattice YY, its pair in the other corpus
- Substitute appropriate arguments from X into the slots of YY

APPROACH USING MSA: EVALUATION

- Corpus:
  - Articles on violence in Palestine and Israel
  - produced between September, 2000 and August 2002
  - by the Agence France-Presse (AFP) and Reuters news agencies
- Ran MSA
- run DIRT on the same dataset

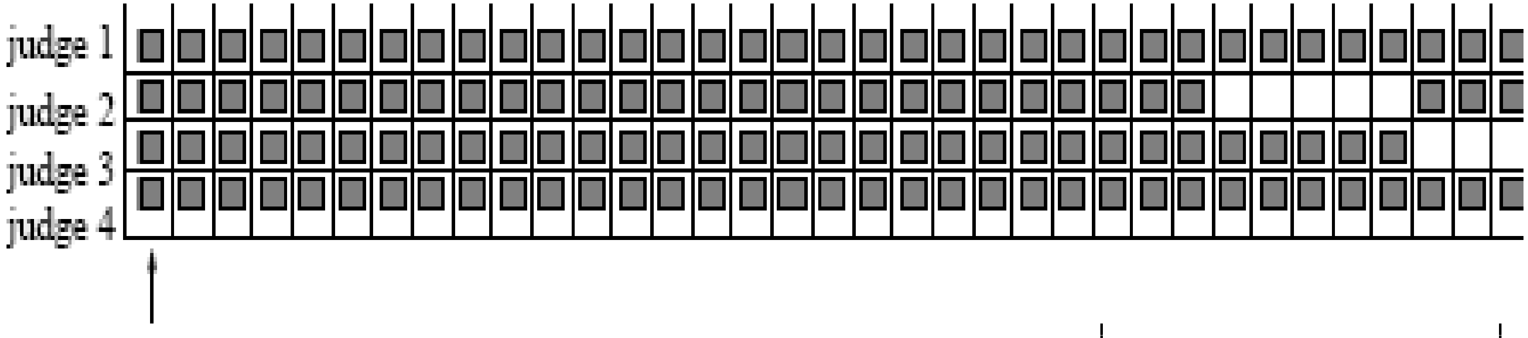
APPROACH USING MSA: EVALUATION MSA VS DIRT

- randomly select 100 pairs of paraphrases from each algorithm's output (50:50)
- 4 judges assessed validity of each pair

- Barzilay and Lee system outperformed DIRT by around 38% points, as rated by 4 judges

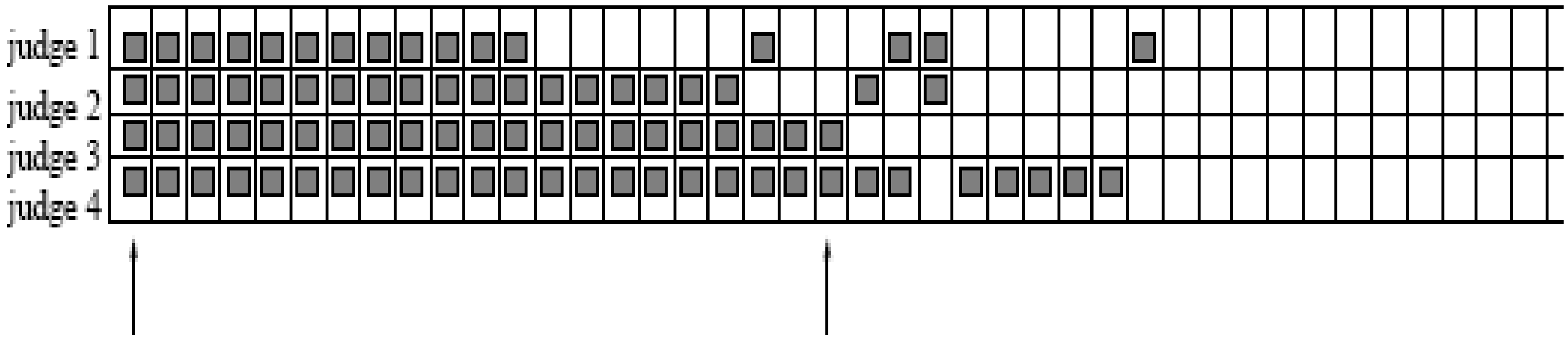
APPROACH USING MSA: EVALUATION MSA VS DIRT

MSA



"DATE: NUM1 are killed and around NUM2 injured when suicide bomber blows up his explosive−packed belt at X1 in X2.";
"palestinian suicide bomber blew himself up at X1 in X2 DATE, killing NUM1 and wounding NUM2 police said."

DIRT (Lin and Pantel)



"X1 forced out of X2"; "X1 ousted from X2" "X1 stormed into X2"; "X1 thrusted into X2"

APPROACH USING MSA: OBSERVATIONS

- mechanism for generating sentence level paraphrases
- unlike some of the previous work which used parallel translations, comparable corpora is used
  - More abundantly available and in many domains
- 80% of the paraphrases have been shown to be accurate

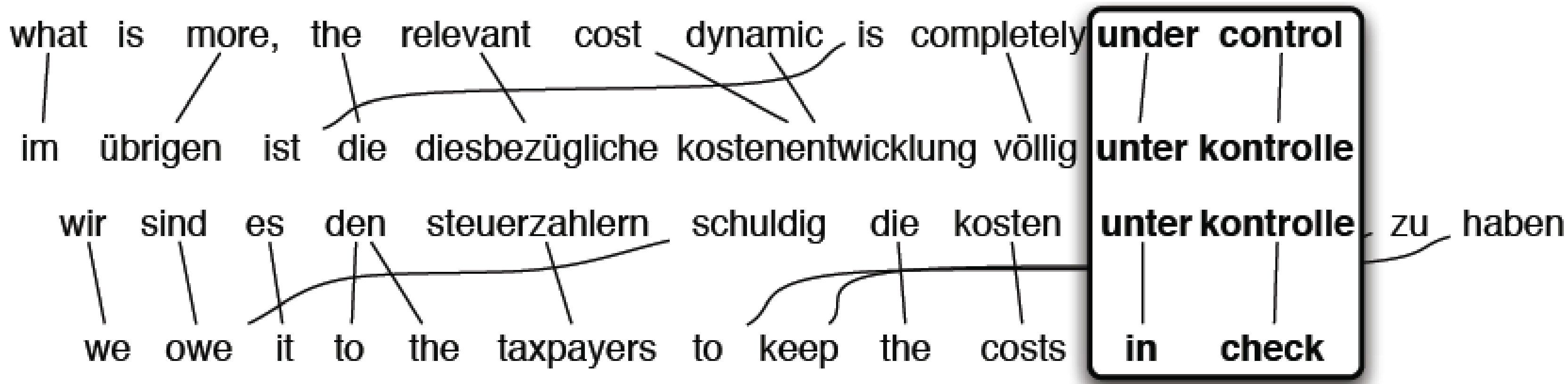
ALGORITHM 3
USING BILINGUAL PARALLEL CORPORA

PARAPHRASING WITH BILINGUAL PARALLEL CORPORA

- C. Bannard & C.Callison-Burch (Edinburgh)
- Idea:
  - identify paraphrases in one language using a phrase in another language as a pivot
  - utilize the abundance of bilingual parallel data

PARAPHRASING WITH BILINGUAL PARALLEL CORPORA

1. look at what foreign language phrases the English translates to
2. find all occurrences of those foreign phrases
3. look back at other English phrases they translate to
4. treat these English phrases as potential paraphrases



PARAPHRASING WITH BILINGUAL PARALLEL CORPORA

- Aligning sentences:
  - aligns phrases by incrementally building longer phrases from words and phrases which have adjacent alignment points
- Find candidate paraphrase e_2

$$
\begin{aligned}
\hat{e_2} &= \arg\max_{e_2 \neq e_1} p(e_2|e_1) \\
&= \arg\max_{e_2 \neq e_1} \sum_f p(f|e_1)p(e_2|f)
\end{aligned}
$$

PARAPHRASING WITH BILINGUAL PARALLEL CORPORA: EVALUATION

Under control
This situation is **in check** in terms of security.
This situation is **checked** in terms of security.
This situation is **curbed** in terms of security.
This situation is **curb** in terms of security.
This situation is **limit** in terms of security.
This situation is **slow down** in terms of security.

- Substitute each set of candidate paraphrases into between 2–10 sentences which contained the original phrase
- 2 judges to check validity of paraphrases:
  - 74.9% - correct paraphrases

PARAPHRASING WITH BILINGUAL PARALLEL CORPORA: OBSERVATIONS

- produces a ranked list of paraphrases with association probabilities
- handles multi-word units
- abundance of bilingual parallel corpora → overcoming domain-specificity issue

Remarks:

- Manually approved alignment – better performance
- What about “lower-density” languages?

CONCLUSIONS

- Algorithms presented:
  - DIRT
  - MSA
  - Paraphrasing with Bilingual Corpora
- Differ with respect to:
  - Unit-length
    - *Phrase vs sentence paraphrases*
  - Data source :
    - *One or several examples of corpora*
    - *Monolingual/bilingual*
    - *Parallel/comparable*
  - Prepossessing:
    - *Parsing*
    - *Sentence alignment*
    - *Clustering*

BIBLIOGRAPHY:

- D. Lin, P. Pantel: Discovery of inference rules for question-answering. Natural Language Engineering 7, 2001.
- Colin Bannard, Chris Callison-Burch: Paraphrasing with bilingual parallel corpora. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. 2005.
- R. Barzilay, L. Lee: Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1.
- http://owl.english.purdue.edu/handouts/research/r_paraphr.html
- http://www.cs.toronto.edu/~gh/Courses/2528/Readings/Barzilay+Lee-slides.pdf