DIRT

Discovery of **I**nference **R**ules from **T**ext

Selected Topics in Semantics and Discourse
Prof. Manfred Pinkal

Mahsa Vafaie

Language and Communication Technologies
University of Saarland

Outline

- Paraphrasing and Inference Rules
- Previous Work
- What is DIRT?
- DIRT Algorithm
- Experimental Results

Problem

“Who is the author of the Star Spangled Banner?”

… Francis Scott Key wrote the “Star Spangled Banner” in 1814.

…comedian-actress Roseanne Barr sang her famous shrieking rendition of the “Star Spangled Banner” before a San Diego Padres-Cincinnati Reds Game.

… Francis Scott Key wrote the “Star Spangled Banner” in 1814.

Paraphrasing and Inference Rules

- How many ways to say “X solves Y?”
  - A paraphrase: “Y is solved by X”
  - An Inference Rule: “X solves Y ≈ X does something about Y”

Inference Rule

Relationships that are not necessarily exact paraphrases, but are nonetheless related and are potentially useful to information retrieval systems.

Inference Rules

- Inference Rules are used in:
  - Question-answering Systems
  - Information Retrieval
  - Summarization
  - …
- Not efficient to create a knowledge base containing Inference Rules manually!

Outline

Previous Work

Most previous works on →

- Natural language generation
- Text summarization
- Information retrieval

Dras M. 1999

Syntactic paraphrase using meta-grammar with synchronous **T**ree **A**djoining **G**rammar (**TAG**) formalism

Barzilay et al. 1999

Multi-document summarization

- → Analyzed 200 two-sentence themes from a corpus
- → Extracted seven lexico-syntactic paraphrasing rules
- → Used to identify common statements in a theme by comparing the predicate-argument structure of the sentences

Richardson 1997

Extracted semantic relationships

- → Hypernym, location, material and purpose
- → Using a parser, constructed a semantic network
- → Used paths in the network to compute the similarity between words

Richardson used paths as features to compute the similarity between words

DIRT uses words as features to compute the similarity of paths

Many text mining algorithms aim to find association rules between terms

DIRT algorithm is set of associations between relations

Outline

What is DIRT?

What is DIRT?

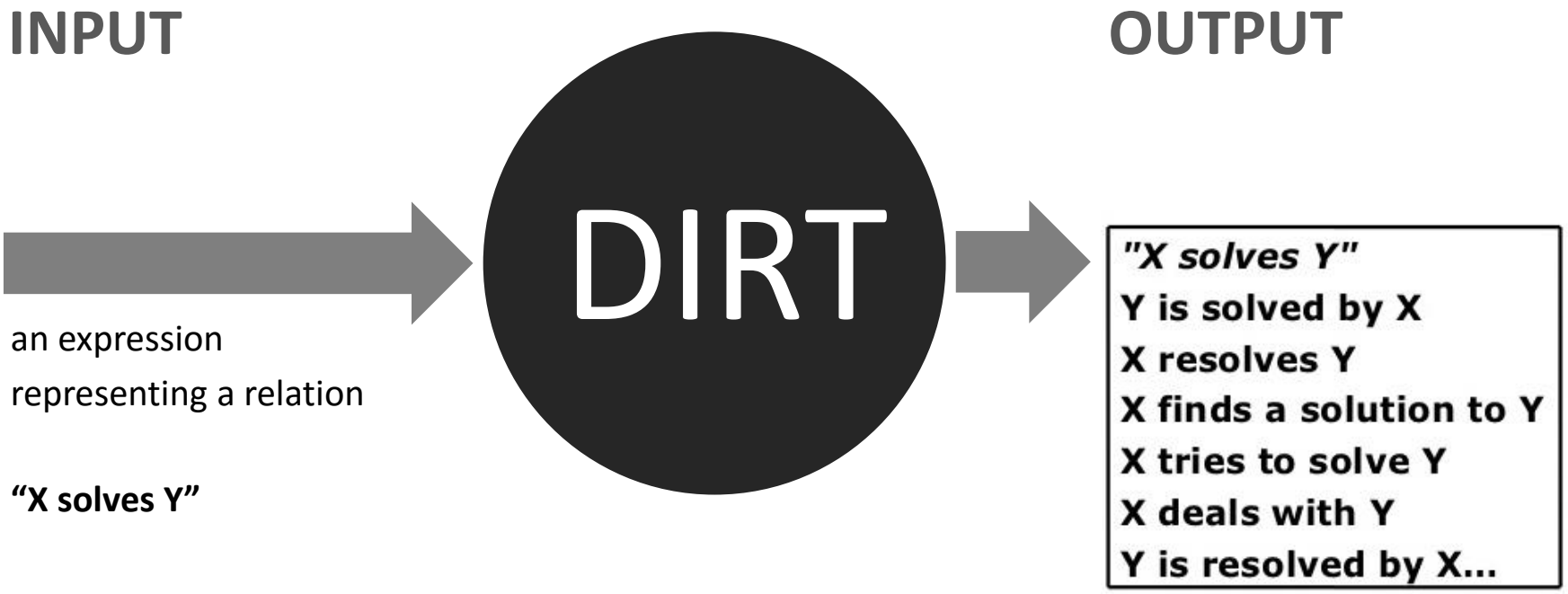
Lin and Pantel in 2001

an unsupervised algorithm aimed at learning expressions that link two nouns and for which a relationship can be inferred

DIRT

an algorithm and a resulting knowledge collection

quite remarkable with 12,000,000 rules!


INPUT
OUTPUT
DIRT
an expression
representing a relation
“X solves Y”
"X solves Y"
Y is solved by X
X resolves Y
X finds a solution to Y
X tries to solve Y
X deals with Y
Y is resolved by X...

Outline

DIRT Algorithm

DIRT Algorithm

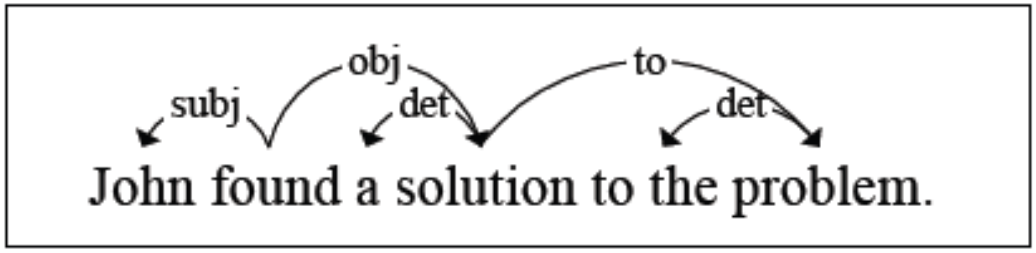
- Dependency Trees
- Paths in Dependency Trees
- The Underlying Assumption of DIRT
- Triples
- Pointwise Mutual Information
- Similarity between paths
- Finding the most similar path

Dependency Trees

Dependency Relationship

An asymmetric binary relationship between a word called **head** and another word called **modifier**

A dependency tree generated by Minipar:



Paths in Dependency Trees

What is a path? →

- a string of links which allows us to represent indirect semantic relationships between two words

How to name a path? →

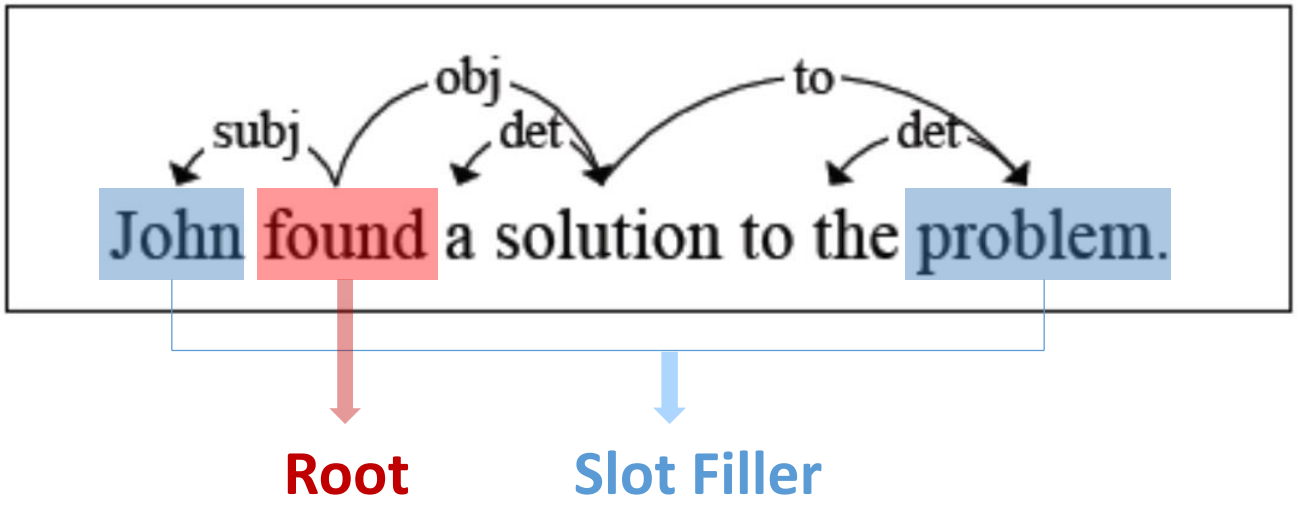
- by concatenating dependency relationships and words along the path excluding the words at the two ends.

The root of the dependency tree does not modify any word. It is called the head of the sentence.

The path between John and problem:

Slot X **Internal Relation** **Slot Y**

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



And the reverse path?

Constraints on the Paths

- Slot fillers must be nouns
- Any dependency relation that does not connect two content words is excluded
- The frequency count of an internal relation must exceed a threshold

Reasons for Constraints on the Paths

- Most meaningful inference rules involve paths that satisfy these conditions
- Reducing the number of distinct paths and thus the amount of computation
- Alleviating the problem of spare data

Example

The paths extracted from this sentence and their meanings are:

(a) N:subj:V←buy→V:from:N
≡ *X* buys something from *Y*

(b) N:subj:V←buy→V:obj:N
≡ *X* buys *Y*

(c) N:subj:V←buy→V:obj:N→sheep→N:nn:N
≡ *X* buys *Y* sheep

(d) N:nn:N←sheep←N:obj:V←buy→V:from:N
≡ *X* sheep is bought from *Y*

(e) N:obj:V←buy→V:from:N
≡ *X* is bought from *Y*

An inverse path is also added for each one above.

Distributional Hypothesis

Words that occur in the same contexts tend to have similar meanings (word similarity computation)

Algorithms Different in:

- Representation of the context:
  1. words in a fixed window
  2. dependency relationships

- Context similarity computation formula

Extended Distributional Hypothesis

Paths that occur in similar contexts, tend to have similar meanings (Discovering Inference Rules)

For each path p that connects two words w_1 and w_2, there are two triples (p, Slot X, w_1) and (p, Slot Y, w_2)

Features of the path

$(\text{Slot X}, w_1)$ $(\text{Slot Y}, w_2)$

The more features two paths share,
The more similar they are

A Triple Database

N:Subj:V ← *pull* → V:Obj:N → *body* → N:from:N

X pulls body from Y:			
SlotX:			
	diver	1	2.45
	equipment	1	1.65
	police	2	2.24
	rescuer	3	4.84
	resident	1	1.60
	who	2	1.32
	worker	1	1.37
SlotY:			
	bus	2	3.09
	coach	1	2.05
	debris	1	2.36
	feet	1	1.75
	hut	1	2.73
	landslide	1	2.39
	metal	1	2.09
	wreckage	3	4.81

Figure 2. An example entry in the triple database for the path "X pulls body from Y".

Pointwise Mutual Information

- PMI measures the degree to which two words are statistically dependent.

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

- Here:

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

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

- Similarity between a pair of slots:

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

- Similarity between a pair of paths

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

Algorithm for finding the most similar paths of *P*

1 Retrieve all the paths that share at least one feature with *P* ➡ **Candidate path**

By storing for each word the set of slots it fills in

2 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%

3 Compute the similarity between *P* and the candidates that passed the filter using *sim* equation and output the paths in descending order of their similarity to *P*

Outline

Experimental Results

Evaluation of the algorithm

by comparing the inference rules it generates

TREC-8 Question Answering Track

TREC (**T**ext **RE**trieval **C**onference)
a U.S. government sponsored competition on information retrieval held annually since 1992

Q#	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?
Q_5	What is the name of the managing director of Apricot Computer?
Q_6	Why did David Koresh ask the FBI for a word processor?

Results

Minipar to parse about 1GB of newspaper text (AP Newswire, San Jose Mercury, and Wall Street Journal)

➡ 7 million paths from the parse trees (231,000 unique) and stored them in a triple database

Results

Q#	PATHS	MAN.	DIRT	INT.	ACC.
Q_1	X is author of Y	7	21	2	52.5%
Q_2	X is monetary value of Y	6	0	0	N/A
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%
Q_5	X is managing director of Y	5	14	1	35.0%
Q_6	X asks Y	2	23	0	57.5%
	asks X for Y	2	14	0	35.0%
	X asks for Y	3	21	3	52.5%

- The paths that we identified from the TREC-8 questions
-
  - manually generated paraphrases of the TREC-8 questions
  - extracted paths from the manually generated paraphrases
  - The number of paths for the manually generated paraphrases of TREC-8 questions
- The number of Top-40 most similar paths by DIRT classified as **correct**
- The intersection between columns three and four
- The percentage of correctly classified paths

Observation

Some of the **correct** paths among the Top-40 extracted by our system for two of the TREC-8 questions

	Q_1	Q_3
PATHS	*X* is author of *Y*	*X* manufactures *Y*
MANUAL VARIATIONS	*Y* is the work of *X*; *X* is the writer of *Y*; *X* penned *Y*; *X* produced *Y*; *X* authored *Y*; *X* chronicled *Y*; *X* wrote *Y*	*X* makes *Y*; *X* produce *Y*; *X* is in *Y* business; *Y* is manufactured by *X*; *Y* is provided by *X*; *Y* is *X*'s product; *Y* is product from *X*; *Y* is *X* product; *Y* is product made by *X*; *Y* is example of *X* product; *X* is manufacturer of *Y*; find *Y* in *X*'s product line; find *Y* in *X* catalog
DIRT VARIATIONS	X co-authors Y; X is co-author of Y; X writes Y; X edits Y; Y is co-authored by X; Y is authored by X; X tells story in Y; X translates Y; X writes in Y; X notes in Y; …	X produces Y; X markets Y; X develops Y; X is supplier of Y; X ships Y; X supplies Y; Y is manufactured by X; X is maker of Y; X introduces Y; X exports Y; X makes Y; X builds Y; X's production of Y; X unveils Y; Y is bought from X; X's line of Y; X assembles Y; X is Y maker; X's Y factory; X's Y production; X is manufacturer of Y; X's Y division; X meets demand for Y; …

DIRT

greatly eases the manual construction of inference rules for an information retrieval system

ANY QUESTIONS

THANKS FOR YOUR ATTENTION

Language and Communication Technologies
University of Saarland