Automatic Acquisition of Paraphrases and Inference Patterns

Nam Khanh Tran

Seminar: Recent Developments in Computational Semantics
Department of Computational Linguistics
Saarland University

May 16, 2011

Outline

1. Introduction
   - Paraphrase
   - Areas of Application
   - Automatic Acquisition of Paraphrases
2. DIRT
   - Paths in Dependency Trees
   - Similarity Measures
   - Finding the Most Similar Paths
   - Experimental Results
3. LEDIR
   - Downside of Automatic Approaches
   - Problem Definition
   - LEDIR Algorithm
   - Experimental Results
4. Conclusion

Outline

Paraphrase

Paraphrases are textual expressions that convey the same meaning using different surface forms

Example

Francis Scotte Key *wrote* the "Star Spangled Banner"
Francis Scotte Key *is the author of* "Star Spangled Banner"
$X\ writes\ Y \Leftrightarrow X\ is\ the\ author\ of\ \ Y$

Areas of Application of Paraphrases

- Question Answering
- Information Retrieval
- Information Extraction
- Text Summarization
- Machine Translation

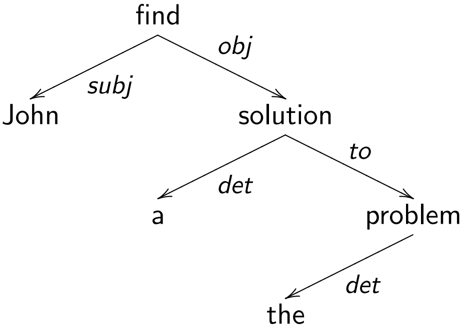
Automatic Acquisition of Paraphrases

- Traditionally knowledge bases are created manually
  - Extremely laborious
  - Difficult to generate a complete list of rules
- General Procedure:
  - Find linguistic structures (= templates) that share the same anchors (= lexical items describing the context in a sentence)
- Automatic Discovery from Text
  - Copus: DIRT [Lin and Pantel, 2001]
  - Web: TE/ASE [Szpektor et al., 2004]

Outline

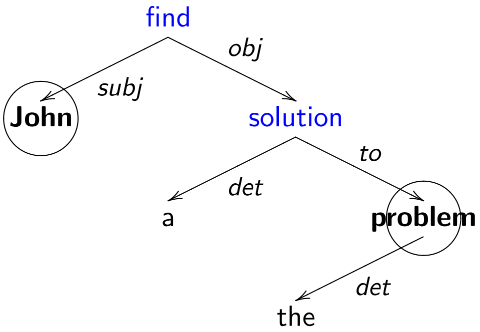
Discovery of Inference Rules from Text (DIRT)

- Discover inference rules between paths in dependency trees
- Dependency trees are generated by an English parser called Minipar



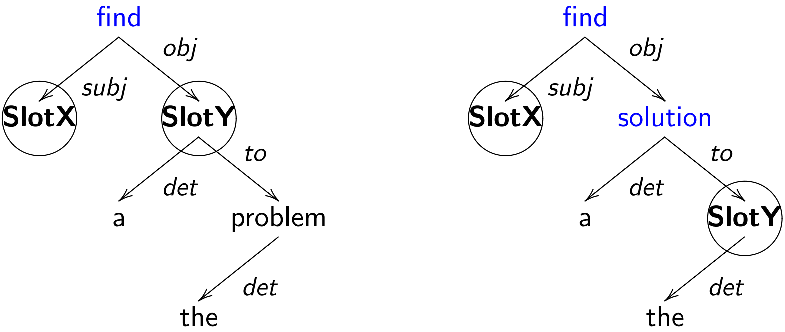
John found a solution to the problem

Paths in Dependency Trees



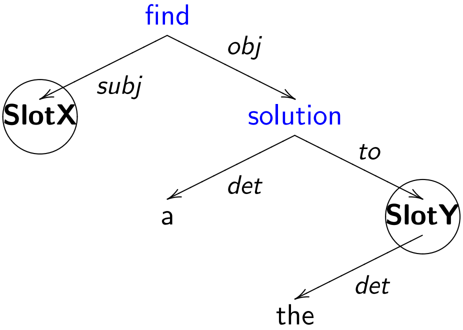
- A path is a concatenation of dependency relationships and words excluding the words at two ends
- A path begins and ends with two dependency relations called *SlotX* and *SlotY*
- The words connected by the path are the fillers of the slots

Paths in Dependency Trees



- Substitute slot fillers by **SlotX** and **SlotY** (e.g: John, solution)
- In a path, dependency relations that are not connected to slots are called **internal relations**.
- A path has to satisfy a set of contraints

Paths in Dependency Trees - Constraints



- Slot fillers must be nouns
- Only consider dependency relations between two content words (i.e, nouns, verbs, adjectives or adverbs)
- The frequency count of an internal relation must exceed a threshold

Assumption

Distributional Hypothesis

Words that occur in the same contexts tend to have similar meanings.

Extended Distributional Hypothesis

If two paths tend to occur in similar contexts, the meanings of the paths tend to be similar.

$\Longrightarrow$ Two paths are similar if their respective **sets of slot fillers** (that occur in a corpus) are similar.

Triple Database

- Collect the frequency counts of all paths and the slot fillers for the paths in the corpus
- For each path p that connects w_1 and w_2 $\Rightarrow$ increase frequency counts of two triples $(p, SlotX, w_1)$ and $(p, SlotY, w_2)$
- $(SlotX, w_1)$ and $(SlotY, w_2)$ are called features of path p $\Rightarrow$ the more features two paths share, the more similar they are

Example

"X finds a solution to Y"

Slot	Slot Filler	Frequency Counts
SlotX	government	2
	he	8
	...	...
SlotY	problem	4
	argument	3
	...	...

Triple Database

- Collect the frequency counts of all paths and the slot fillers for the paths in the corpus
- For each path p that connects w_1 and w_2 $\Rightarrow$ increase frequency counts of two triples $(p, SlotX, w_1)$ and $(p, SlotY, w_2)$
- $(SlotX, w_1)$ and $(SlotY, w_2)$ are called features of path p $\Rightarrow$ the more features two paths share, the more similar they are

Example

"X finds a solution to Y"

Slot	Slot Filler	Frequency Counts
SlotX	government	2
	he	8
	...	...
SlotY	problem	4
	argument	3
	...	...

Problem?

Mutual Information between Path, Slot and Slot Filler

- Compute the mutual information between all pairs of paths and slot fillers
- Measure strength of the association between a slot and a filler

Mutual Information between Path, Slot and Slot Filler

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

Mutual Information between Path, Slot, Slot Filler

Mutual Information between Path, Slot and Slot Filler

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

$|p, Slot, w|$ = frequency count of the triple $(p, Slot, w)$

$$|p, Slot, *| = \sum_{w} |p, Slot, w| \qquad |*, *, *| = \sum_{p,s,w} |p, s, w|$$

Mutual Information between Path, Slot and Slot Filler

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

Mutual Information between Path, Slot, Slot Filler

Mutual Information between Path, Slot and Slot Filler

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

$|p, Slot, w|$ = frequency count of the triple $(p, Slot, w)$

$|p, Slot, *| = \sum_{w} |p, Slot, w|$ $\qquad |*, *, *| = \sum_{p,s,w} |p, s, w|$

Mutual Information between Path, Slot and Slot Filler

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

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

Triple Database

Example

X finds a solution to Y

Slot	Slot Filler	Frequency Counts	Mutual Information
SlotX	government	2	3.14
	he	8	1.23
	president	3	2.48
	...	...	...
SlotY	problem	4	4.15
	argument	3	2.27
	issue	2	2.19
	...	...	...

Similarity between a Pair of Slots

Slot Similarity

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

$slot_1 = (p_1, s)$
$slot_2 = (p_2, s)$
$T(p_i, s) =$ set of words that fill in the s slot of path p_i

Similarity between a Pair of Paths

Path Similarity

Similarity between two paths p_1 and p_2

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

Finding the Most Similar Paths

- Large number of paths in the triple database
  $\rightarrow$ Computing the similarity between every pair of paths is impractical
- Algorithm for finding the most similar paths of p
  1. Retrieve all the paths that share at least one feature with p
     $\rightarrow$ candidate paths
  2. For each candidate path c, count the number of features shared by c and p, filter out c if the number of common features is too small
  3. Compute similarity between p and c $\longrightarrow$ output (ranked list)

Example

X solves Y: X resolves Y, X finds a solution to Y, X deals with Y, X tackles Y, ...

Experimental Results

- Compare with a set of human-generated paraphrases on 6 questions in TREC-8 Question-Answering Track.
- Perform DIRT algorithm on 1GB of newspaper text $\rightarrow$ 7 millions paths
- Manually inspect the top 40 outputs of each input path (correct/incorrect)

Experimental Results

First six questions from TREC-8

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?

Experimental Results

Evaluation of Top-40 most similar paths

Q	Paths	Human	DIRT	Accuracy
Q_1	X is author of Y	7	21	52.5%
Q_2	X is monetary value of Y	6	0	N/A
Q_3	X manufactures Y	13	37	92.5%
Q_4	X spend Y	7	16	40.0%
	spend X on Y	8	15	37.5%
Q_5	X is managing director of Y	5	14	35.0%
Q_6	X asks Y	2	23	57.5%
	asks X for Y	2	14	35.0%
	X asks for Y	3	21	52.5%

Experimental Results

Observations:

- Little overlap between manually generated and machine generated phrases ⇒ Paraphrase generation is difficult both for humans and machines.
- DIRT outputs: Humans can easily identify correct phrases ⇒ DIRT can help humans to build paraphrase knowledge bases

Problems:

- "X worsens Y" has a high similarity to "X solves Y"
- All rules are considered symmetric ("X eats Y" ⇔ "X likes Y") ⇒ not really true

Experimental Results

Observations:

- Little overlap between manually generated and machine generated phrases ⇒ Paraphrase generation is difficult both for humans and machines.
- DIRT outputs: Humans can easily identify correct phrases ⇒ DIRT can help humans to build paraphrase knowledge bases

Problems:

- "X worsens Y" has a high similarity to "X solves Y"
- All rules are considered symmetric ("X eats Y" ⇔ "X likes Y") ⇒ not really true

LEarning Directionality of Inference Rules!

Outline

1. Introduction
   - Paraphrase
   - Areas of Application
   - Automatic Acquisition of Paraphrases
2. DIRT
   - Paths in Dependency Trees
   - Similarity Measures
   - Finding the Most Similar Paths
   - Experimental Results
3. LEDIR
   - Downside of Automatic Approaches
   - Problem Definition
   - LEDIR Algorithm
   - Experimental Results
4. Conclusion

Downside of Automatic Approaches

Inference rules are underspecified in directionality

X eats Y $\Leftrightarrow$ X likes Y

John eats *spicy food* $\Rightarrow$ *John* likes *spicy food*

John likes *rollerblading* $\not\Rightarrow$ *John* eats *rollerblading*

Downside of Automatic Approaches

Large amount of incorrect inference rules

X is charged by Y $\Rightarrow$ Y announced the arrest of X

Nichols was charged by *federal prosecutors* for murder
$\Rightarrow$ *Federal prosecutors* announced the arrest of *Nichols*

Accounts were charged by *CCM telemarketers* without obtaining authorizations
$\not\Rightarrow$ *CCM telemarketers* announced the arrest of *accounts*

Problem Definition

Goal: Filter out incorrect inference rules and identify the directionality of the correct ones

Formally

Given the inference rule $p_i \Leftrightarrow p_j$, we want to conclude which one of the following is more appropriate:

1. $p_i \Leftrightarrow p_j$
2. $p_i \Rightarrow p_j$
3. $p_i \Leftarrow p_j$
4. No plausible inference

Assumption

Distributional Hypothesis

Words that occur in the same contexts tend to have similar meanings.

Assumption

Distributional Hypothesis

Words that occur in the same contexts tend to have similar meanings.

Directionality Hypothesis

If two binary semantic relations tend to occur in similar contexts and the first one occurs in significantly more contexts than the second, then the second most likely implies the first and not vice versa.

Example

There are many more things that someone might like than those that someone might eat $\rightarrow$ *"X eats Y"* $\Rightarrow$ *"X likes Y"*

Steps of the Algorithm

Given a candidate inference rule $p_i \Leftrightarrow p_j$:

1. Model the contexts of p_i and p_j by selectional preferences
2. Determine the plausibility of the inference rule
3. If it is plausible, determine its directionality

Model the contexts of a relation

Let $\langle x, p, y \rangle$ be an instance of the relation p
Let C_x and C_y be the semantic classes of the words that can be instantiated for x and y

Example

X is charged by Y
$C_x = \{social\_group, organism, state, ...\}$
$C_y = \{authority, state, section, ...\}$

Joint Relational Model (JRM)

Given a relation p and a large corpus of (English) text:

1. Find all occurrences of relation p
2. For every instance $\langle x, p, y \rangle$
   - Obtain the sets C_x and C_y of the semantic classes that x and y belong to
   - Every triple $\langle c_x, p, c_y \rangle$ is a candidate selectional preference for p, by assuming that every $c_x \in C_x$ can co-occur with every $c_y \in C_y$ and vice versa
3. Rank these candidates using Pointwise mutual information

Joint Realtional Model (JRM)

Ranking candidates

The ranking function is defined as the strength of association between two semantic classes c_x and c_y

Pointwise mutual information

$$pmi(c_x|p; c_y|p) = log \frac{P(c_x, c_y|p)}{P(c_x|p)P(c_y|p)}$$

Joint Realtional Model (JRM)

Ranking candidates

Maximum likelihood estimates over the corpus

$$P(c_x|p) = \frac{|c_x, p, *|}{|*, p, *|} \quad P(c_y|p) = \frac{|c_y, p, *|}{|*, p, *|} \quad P(c_x, c_y|p) = \frac{|c_x, p, c_y|}{|*, p, *|}$$

$$|c_x, p, *| = \sum_{w \in c_x} \frac{|w, p, *|}{|C(w)|} \quad |*, p, c_y| = \sum_{w \in c_y} \frac{|*, p, w|}{|C(w)|}$$

$$|c_x, p, c_y| = \sum_{w_1 \in c_x, w_2 \in c_y} \frac{|w_1, p, w_2|}{|C(w_1) \times C(w_2)|}$$

$|c_x, p, c_y|$: frequency of observing instance $\langle c_x, p, c_y \rangle$
$|x, p, y|$: frequency of observing instance $\langle x, p, y \rangle$
$|C(w)|$: number of classes to which w belongs

Independent Relational Model (IRM)

Given a relation p and a large corpus of (English) text

1. Find all occurrences of relation p
2. For each instance $\langle x, p, y \rangle$:
    - Obtain the sets C_x and C_y of semantic classes that x and y belong to
    - All triples $\langle c_x, p, * \rangle$ and $\langle *, p, c_y \rangle$ are independent candidate selectional preferences for p, where $c_x \in C_x$ and $c_y \in C_y$
3. Rank candidates by using maximum likelihood estimates for $\mathrm{P}(c_x|p)$ and $\mathrm{P}(c_y|p)$

Indepedent Relational Model (IRM)

Convert independently learned candidates into a joint representation for use by the inference plausibility and directionality model

Joint Representation

Cartesian product of sets $\langle C_x, p, *\rangle$ and $\langle *, p, C_y\rangle$

$$\langle C_x, p, *\rangle \times \langle *, p, C_y\rangle = \left\{ \begin{array}{c} \langle c_x, p, c_y\rangle : \forall\langle c_x, p, *\rangle \in \langle C_x, p, *\rangle \quad and \\ \forall\langle *, p, c_y\rangle \in \langle *, p, C_y\rangle \end{array} \right\}$$

Inference plausibility

Overlap coefficient between two vectors A and B

$$sim(A,B) = \frac{|A \cap B|}{min(|A|,|B|)}$$

Overlap coefficient between the selectional preferences of p_i and p_j

$$sim(p_i,p_j) = \frac{|\langle C_x, p_i, C_y\rangle \cap \langle C_x, p_j, C_y\rangle|}{min(|\langle C_x, p_i, C_y\rangle|, |\langle C_x, p_j, C_y\rangle|)}$$

Given a candidate inference rule $p_i \Leftrightarrow p_j$ and the respective selectional preferences:

$$\begin{aligned}&\textit{If}\ \ sim(p_i,p_j) \geq \alpha:\\ &\qquad \textit{the inference is plausible}\\ &\textit{else}:\\ &\qquad \textit{the inference is not plausible}\end{aligned}$$

Directionality model

For a plausible inference:

If	$\frac{\lvert C_x, p_i, C_y \rvert}{\lvert C_x, p_j, C_y \rvert} \geq \beta$	we conclude $p_i \Leftarrow p_j$
else if	$\frac{\lvert C_x, p_i, C_y \rvert}{\lvert C_x, p_j, C_y \rvert} \leq \frac{1}{\beta}$	we conclude $p_i \Rightarrow p_j$
else		we conclude $p_i \Leftrightarrow p_j$

$\beta \geq 1$

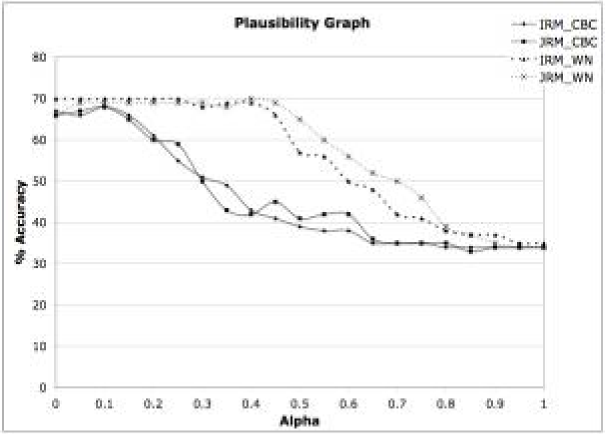
Experimental setup

- Inference rules from DIRT resource
- Two sets of semantic classes:
  - 1628 semantic classes obtained by running the CBC clustering algorithm on newswire collections
  - 1287 semantic classes from WordNet synsets at depth four
- 1999 AP newswire collection (31 million words)
- Manually annotated **gold standard**:
  - 57 DIRT inference rules
  - The most appropriate of four tags ($\Rightarrow$ / $\Leftarrow$ / $\Leftrightarrow$ / NO) is assigned to inference rule

Results

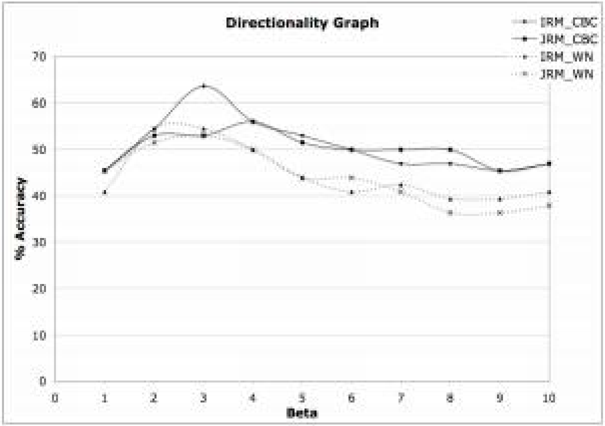
Model		α	β	Accuracy (%)
B-random		-	-	25
B-frequent		-	-	34
B-DIRT		-	-	25
JRM	CBC	0.15	2	38
	WN	0.15	2	38
IRM	**CBC**	**0.15**	**3**	**48**
	WN	0.45	2	43

Results



Accuracy variation in predicting correct versus incorrect inference rules for different values of α

Results



Accurary variation in predicting directionality of correct inference rules for different values β

Outline

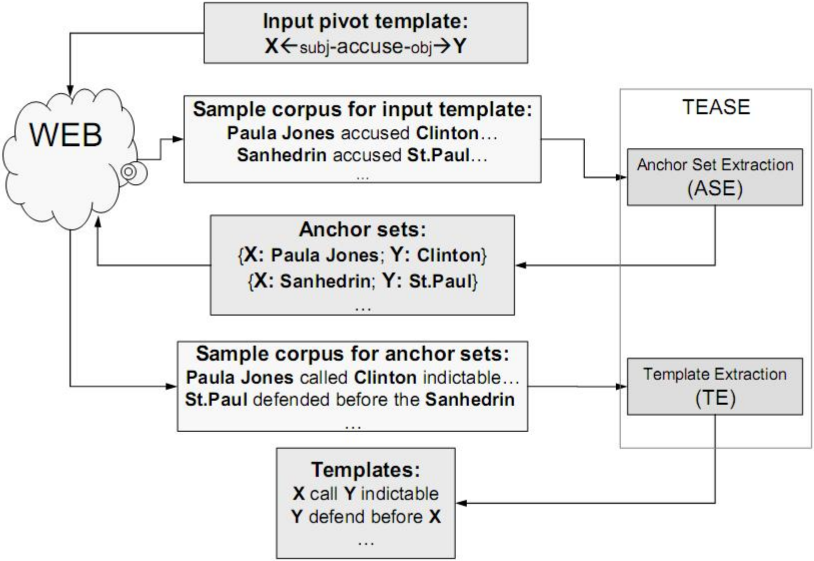
Conclusion

- DIRT: learns paraphrase patterns by computing similarity between slots of dependency paths
  $\longrightarrow$ Methods to learn templates with an arbitrary number of slots?
- LEDIR: filters incorrect inference rules and identifies the directionality of the correct ones by using selectional preferences
  $\longrightarrow$ Antonymy relations like "*X loves Y*" $\Leftrightarrow$ "*X hates Y*"?

References

- D. Lin and P. Pantel. DIRT: Discovery of Inference Rules from Text. Proceedings of ACM Conference on Knowledge Discovery and Data 2001.
- Bhagat, R., Pantel, P., and Hovy, E. LEDIR: An Unsupervised Algorithm for Learning Directionality of Inference Rules. Proceedings of EMNLP-CoNLL 2007.
- Pantel, P., Bhagat, R., Coppola, B., Chklovski, T., and Hovy, E. ISP: Learning Inferential Selectional Preferences. Proceedings of NAACL/HLT 2007.

Thank you for your attention!


Input pivot template:
X←subj-accuse-obj→Y
WEB
Sample corpus for input template:
Paula Jones accused Clinton...
Sanhedrin accused St.Paul...
...
TEASE
Anchor Set Extraction
(ASE)
Anchor sets:
{X: Paula Jones; Y: Clinton}
{X: Sanhedrin; Y: St.Paul}
...
Sample corpus for anchor sets:
Paula Jones called Clinton indictable...
St.Paul defended before the Sanhedrin
...
Template Extraction
(TE)
Templates:
X call Y indictable
Y defend before X
...