J.H. Martin: Computer Understanding of Conventional Metaphoric Language

Computational Approaches to Creative Language (SS 2010)

Tassilo Barth

Saarland University

01. June 2010

Conventional Metaphors

- How can I **kill** a process?
- How can I **enter** Emacs?
- Inflation is **eating** up our savings.
- He **stole** my time.

Conventional Metaphors

- How can I **kill** a process?
- How can I **enter** Emacs?
- Inflation is **eating** up our savings.
- He **stole** my time.

Overview

Former approaches

- Word-sense approach: Store metaphoric meanings as separate word senses
- Similarity approach: Complex search process, looking for similarities between domains
- Acc. to Martin: “Knowledge deficient“ approaches

The Metaphoric Knowledge Approach

Specific **knowledge** about conventional metaphors, applied to..
.. acquisition, use and representation.

Based on a knowledge base of existing metaphors and their relations

Based on Psycholinguistic findings

- **Total processing time**: Similar mechanisms for literal and metaphorical interpretations
- **Non-Optionality** of metaphorical interpretations

The Computational System

- MIDAS: **M**etaphor **I**nterpretation, **D**enotation and **A**quisition **S**ystem
- Applied to "Unix Consultant"
- Representation, Interpretation and Acquisition

How?

- KODIAK formalism: representation of semantic networks
- Concepts connected by links

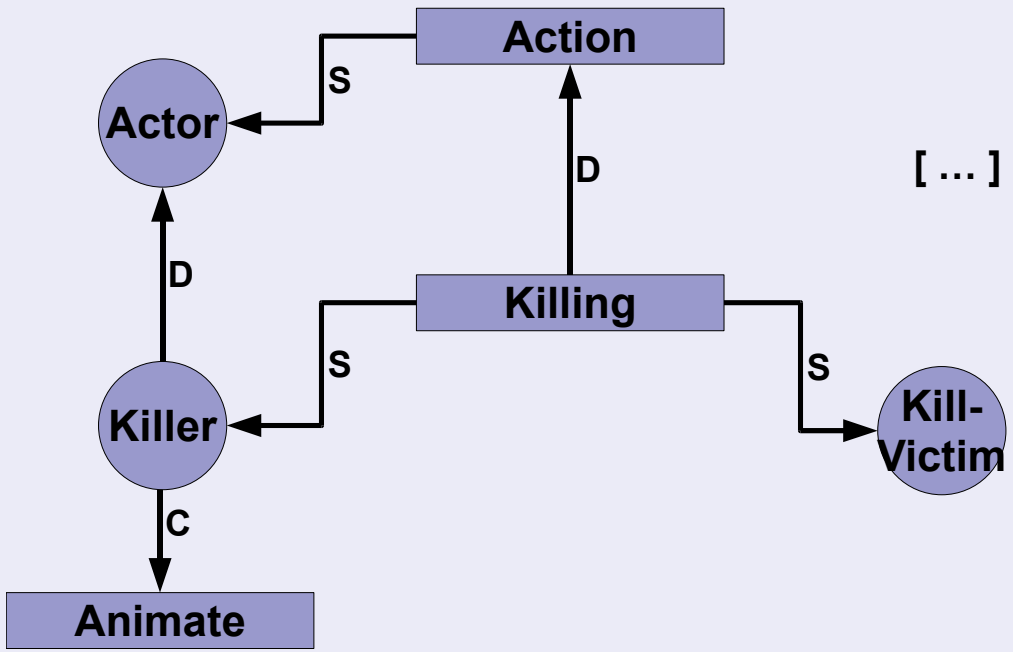
Building blocks

- Source concepts

Building blocks

- Source concepts

Source concept: Killing (shortened)



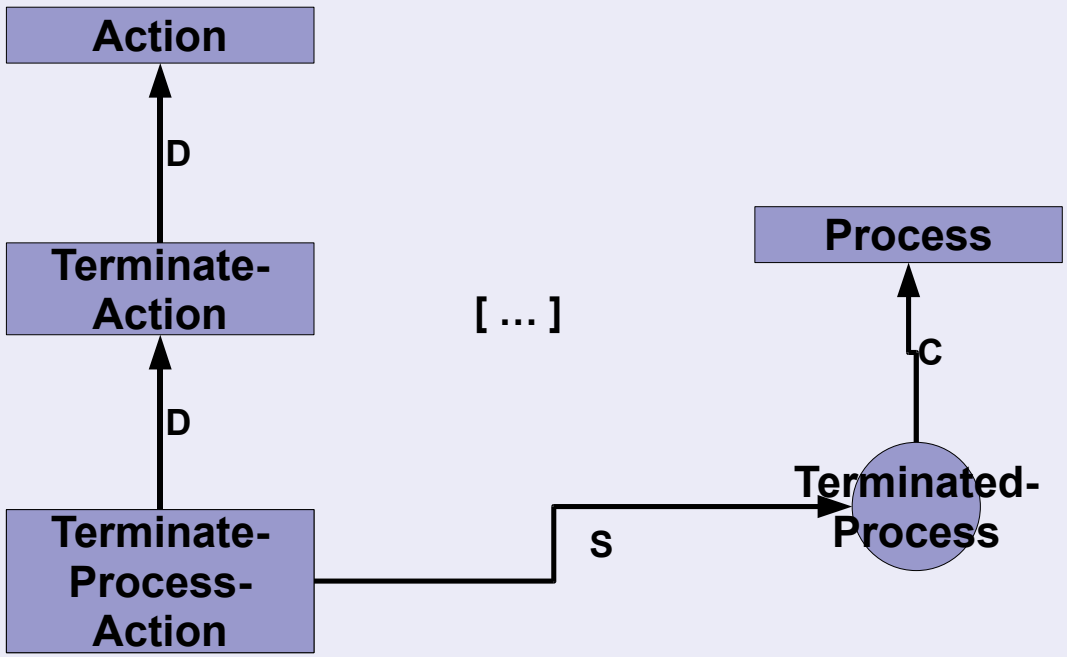
Building blocks

- Source concepts and target concepts

Building blocks

- Source concepts and target concepts

Target concept: Terminate-Process-Action (shortened)



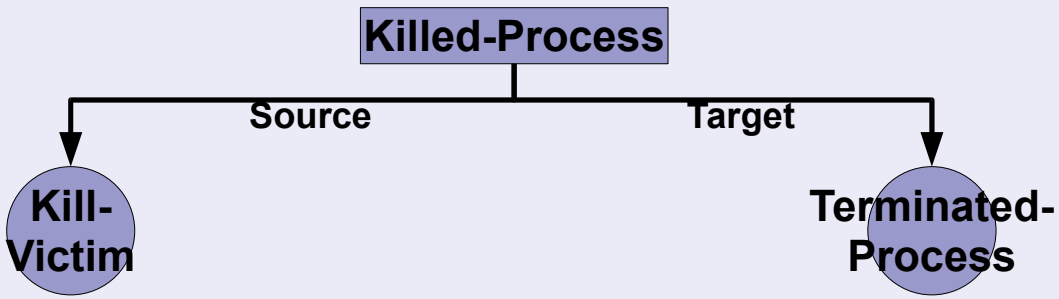
Building blocks

- Source concepts and target concepts
- Connected by metaphor maps

Building blocks

- Source concepts and target concepts
- Connected by metaphor maps

Metaphor map: Killed-Process (shortened)



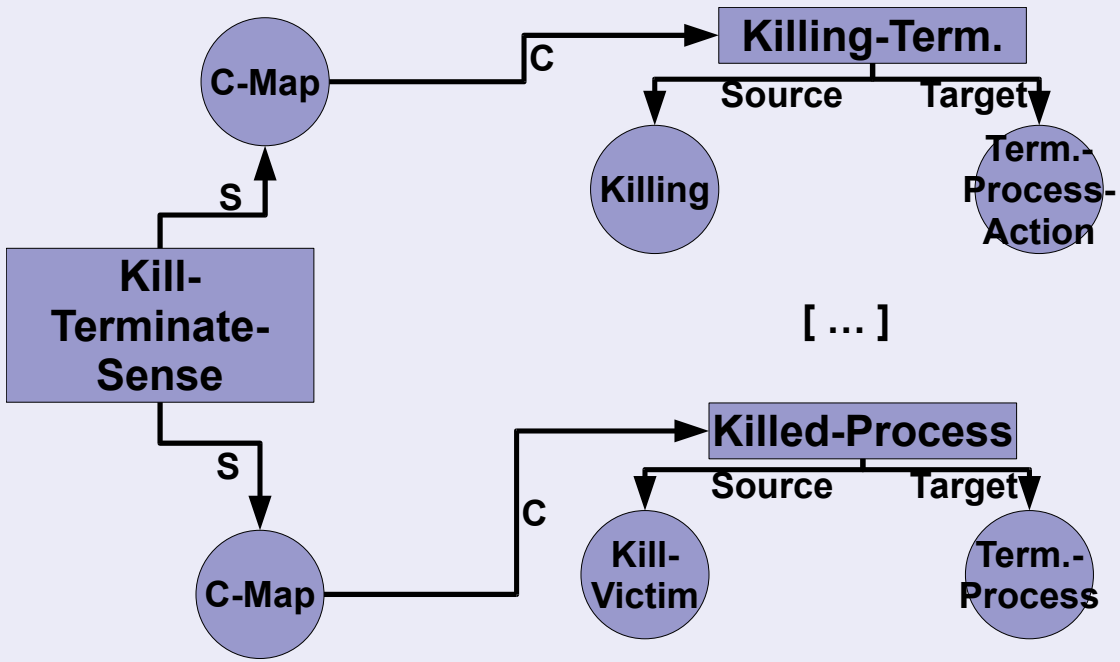
Building blocks

- Source concepts and target concepts
- Connected by metaphor maps
- Many of which form a metaphor sense

Building blocks

- Source concepts and target concepts
- Connected by metaphor maps
- Many of which form a metaphor sense

Metaphor sense: Kill-Terminate (shortened)



Generalization

- Ex. “How can I get into Emacs?”, “How can I enter into a security enabled wireless network?”
- Generalization:

Generalization

- Ex. "How can I get into Emacs?", "How can I enter into a security enabled wireless network?"
- Generalization:

Abstract metaphor Enclosing-Participating

"Enter an Enclosure" $\longrightarrow$ "Participate in an activity"

Generalization

- Ex. "How can I get into Emacs?", "How can I enter into a security enabled wireless network?"
- Generalization:

Concrete metaphor Enclosed-Using-Computer-Process

"Enter an Enclosure" $\longrightarrow$ "Participate/Start a computer process / a wlan network"

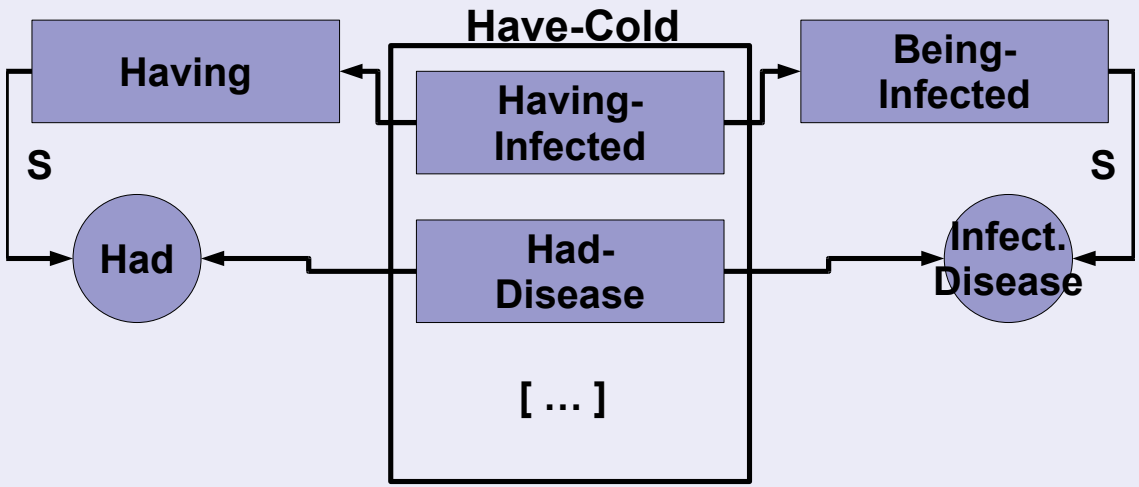
Similarities between Metaphors

- Ex. “John gave Mary a cold” and “Mary has a cold”

Similarities between Metaphors

- Ex. "John gave Mary a cold" and "Mary has a cold"
- Relationship between Have-Cold

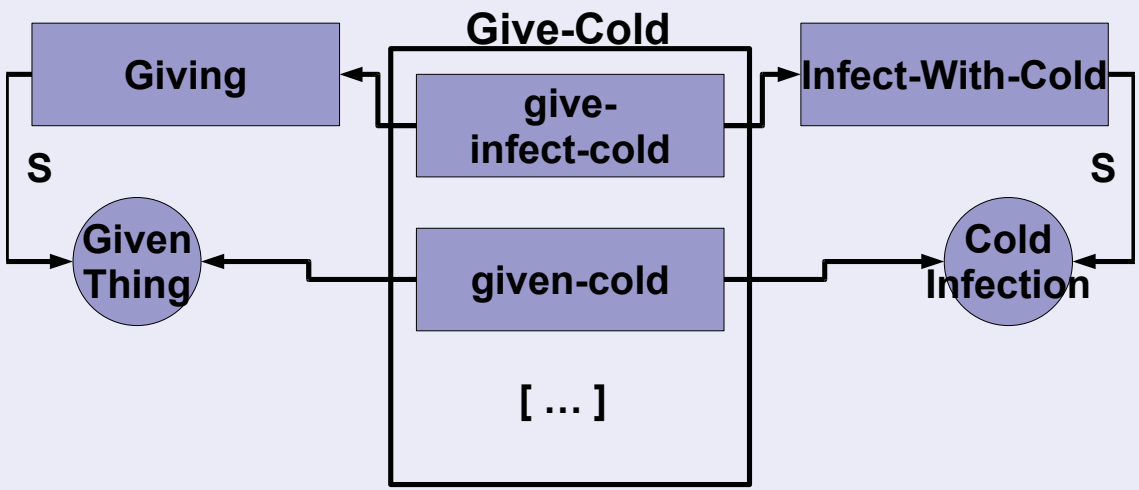
Metaphor sense: Have-Cold (shortened)



Similarities between Metaphors

- Ex. "John gave Mary a cold" and "Mary has a cold"
- Relationship between Have-Cold
- .. and Give-Cold

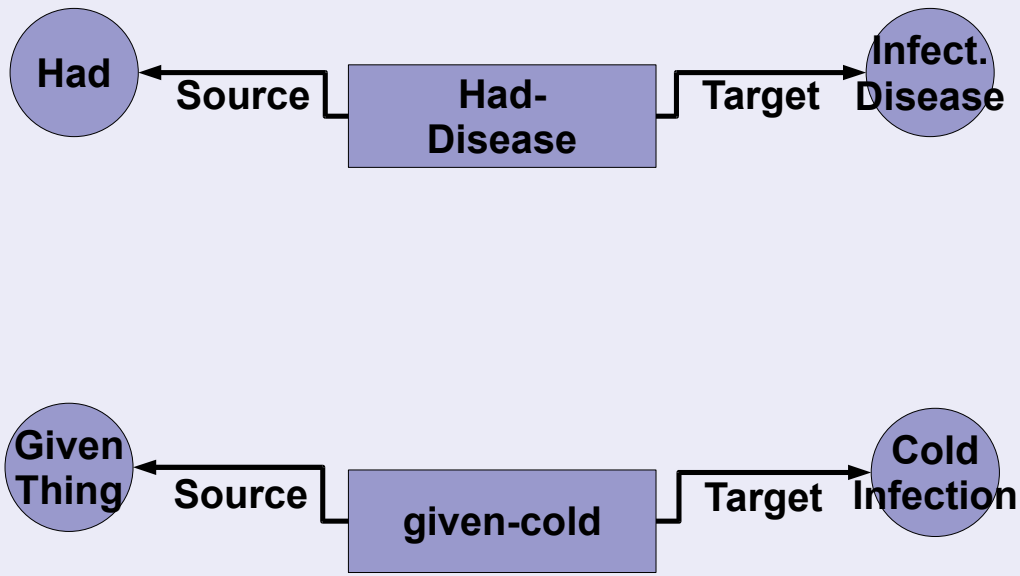
Metaphor sense: Give-Cold (shortened)



Similarities between Metaphors

- Ex. "John gave Mary a cold" and "Mary has a cold"
- Relationship between Have-Cold
- .. and Give-Cold
- Give-Cold and Have-Cold share Metaphor Maps!

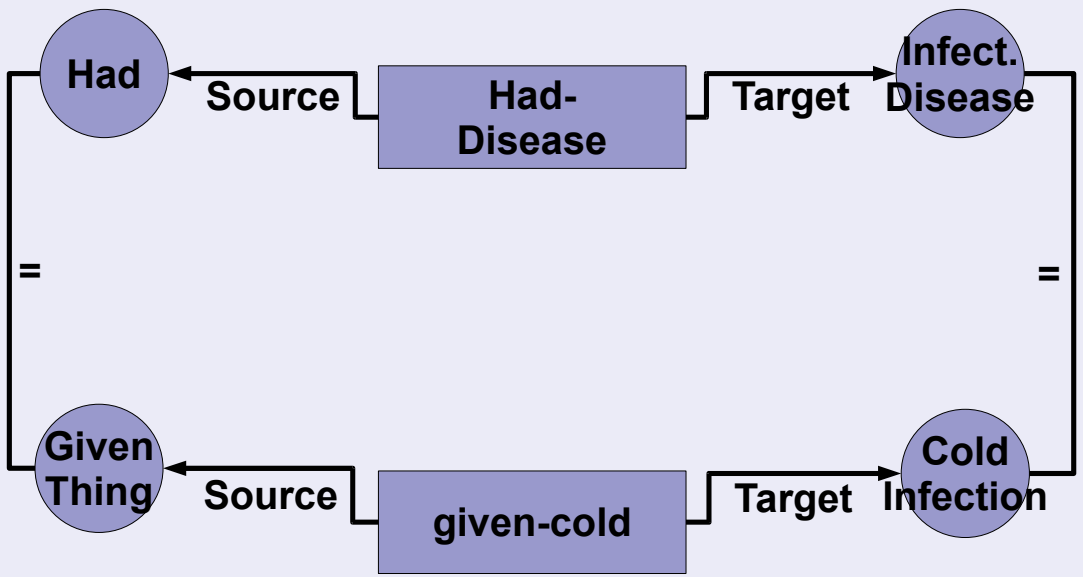
Ex. Relationship between Have- and Give-Cold



Similarities between Metaphors

- Ex. "John gave Mary a cold" and "Mary has a cold"
- Relationship between Have-Cold
- .. and Give-Cold
- Give-Cold and Have-Cold share Metaphor Maps!

Ex. Relationship between Have- and Give-Cold



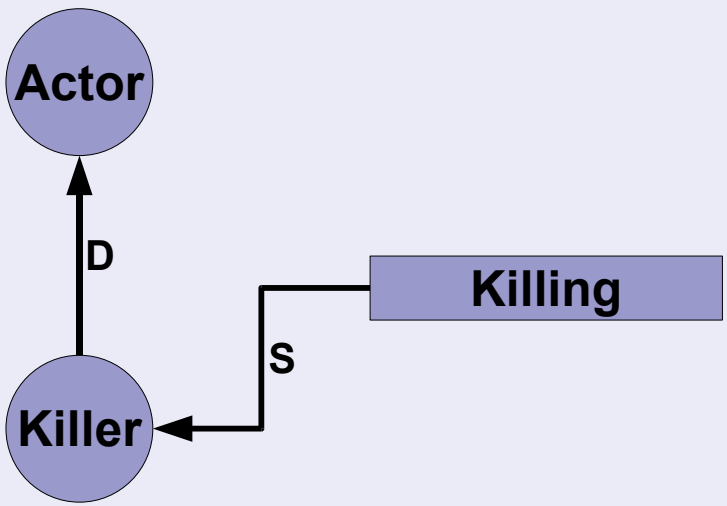
Core mechanisms

- Concretion

Core mechanisms

- Concretion

Concretion example



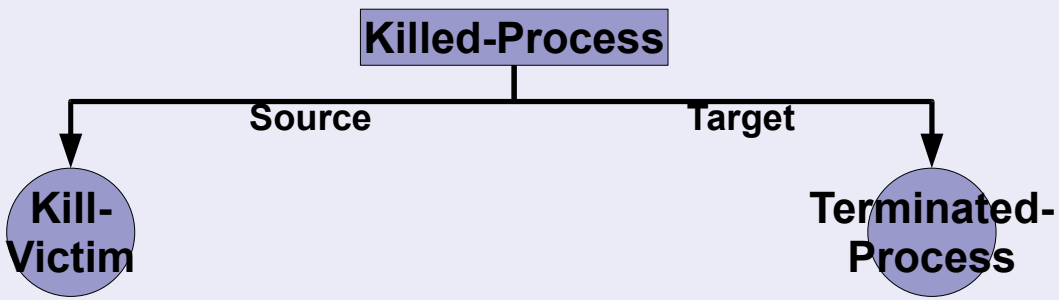
Core mechanisms

- Concretion
- Metaphorical Unviewing

Core mechanisms

- Concretion
- Metaphorical Unviewing

Unviewing example



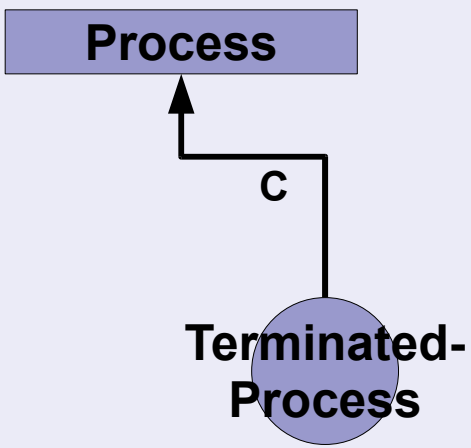
Core mechanisms

- Concretion
- Metaphorical Unviewing
- Both: Constraint Checking

Core mechanisms

- Concretion
- Metaphorical Unviewing
- Both: Constraint Checking

Constraint checking example



Algorithm

1. **Initial parse:** Primal representation

Algorithm

1. **Initial parse:** Primal representation

"How can I kill a process?"

(A Killing1 (↑ Killing)
(agent123 (↑ agent) (A I11 (↑ I)))
(patient321 (↑ patient) (A process12 (↑ process)))

Algorithm

1. **Initial parse:** Primal representation
2. Case roles **concreted** to actual concepts

Algorithm

1. **Initial parse:** Primal representation
2. Case roles **concreted** to actual concepts

Concretion has happened

(A Killing1 (↑ Killing)
(killer1 (↑ killer) (A I11 (↑ I)))
(kill-victim1 (↑ kill-victim) (A process12 (↑ process)))

Algorithm

1. **Initial parse:** Primal representation
2. Case roles **concreted** to actual concepts
3. **Collection** of all interpretations (Using Unviewing + Concretion)

Algorithm

1. **Initial parse:** Primal representation
2. Case roles **concreted** to actual concepts
3. **Collection** of all interpretations (Using Unviewing + Concretion)

One literal, one metaphorical interpretation

Killing1 as Killing.
Killing1 as Killing-Terminate.

Algorithm

1. **Initial parse:** Primal representation
2. Case roles **concreted** to actual concepts
3. **Collection** of all interpretations (Using Unviewing + Concretion)
4. **Validation**: check semantic constraints

Algorithm

1. **Initial parse:** Primal representation
2. Case roles **concreted** to actual concepts
3. **Collection** of all interpretations (Using Unviewing + Concretion)
4. **Validation**: check semantic constraints

Literal fail, metaphorical win

Failed interpretation: Killing1 as Killing.
Valid metaphorical interpretation.

Algorithm

1. **Initial parse:** Primal representation
2. Case roles **concreted** to actual concepts
3. **Collection** of all interpretations (Using Unviewing + Concretion)
4. **Validation**: check semantic constraints
5. **Application** (followed by further Concretion if necessary)

Algorithm

1. **Initial parse:** Primal representation
2. Case roles **concreted** to actual concepts
3. **Collection** of all interpretations (Using Unviewing + Concretion)
4. **Validation**: check semantic constraints
5. **Application** (followed by further Concretion if necessary)

Applying conventional metaphor Killing-Terminate.

(A Killing-Terminate1 (↑ Killing-Terminate)
(agent-of-termination1
(↑ agent-of-termination) (A I11 (↑ I)))
(terminated-process1
(↑ terminated-process) (A process12 (↑ process)))

Algorithm

1. **Initial parse:** Primal representation
2. Case roles **concreted** to actual concepts
3. **Collection** of all interpretations (Using Unviewing + Concretion)
4. **Validation**: check semantic constraints
5. **Application** (followed by further Concretion if necessary)
6. **Return**: list of resulting interpretations. > 1 if input ambiguous

Algorithm

1. **Initial parse:** Primal representation
2. Case roles **concreted** to actual concepts
3. **Collection** of all interpretations (Using Unviewing + Concretion)
4. **Validation**: check semantic constraints
5. **Application** (followed by further Concretion if necessary)
6. **Return**: list of resulting interpretations. > 1 if input ambiguous

Final interpretation: "How can I kill a process?"

(A How-Q207 (↑ How-Q)
(topic208 (↑ topic)
(A Killing-Terminate1 (↑ Killing-Terminate)
(agent-of-termination1
(↑ agent-of-termination) (A I11 (↑ I)))
(terminated-process1
(↑ terminated-process) (A process12 (↑ process))))))

Learning new metaphors

- Unknown metaphors: Extend known metaphors using **analogy operations**

Learning new metaphors

- Unknown metaphors: Extend known metaphors using **analogy operations**
- Ex.: "How can I kill a conversation?"
  No known interpretation?

Learning new metaphors

- Unknown metaphors: Extend known metaphors using **analogy operations**
- Ex.: "How can I kill a conversation?"
  No known interpretation?
- Conversation is-a Process

Learning new metaphors

- Unknown metaphors: Extend known metaphors using **analogy operations**
- Ex.: "How can I kill a conversation?"
  No known interpretation?
- Conversation is-a Process
- Known metaphor: Kill-process - kill = terminate a process

Learning new metaphors

- Unknown metaphors: Extend known metaphors using **analogy operations**
- Ex.: "How can I kill a conversation?"
  No known interpretation?
- Conversation is-a Process
- Known metaphor: Kill-process - kill = terminate a process
- New metaphor: Kill-conversation

Conclusion - Metaphor Knowledge Approach/MIDAS

- Unites advantages of word-sense and similarity approach
- In accord with some psycholinguistic findings
- Represents relations between metaphors
- Model for: Representation, Interpretation, Acquisition

But..

Requirement: predefined knowledge base of conventional metaphors

- Origin of the initial knowledge?
- Which size?
- Success (Recall)?

But..

Requirement: predefined knowledge base of conventional metaphors

- Origin of the initial knowledge?
- Which size?
- Success (Recall)?

- And how can I kill a process, after all?

The answer is Ctrl + C!

Bibliography:

- Martin, J.H. (1992). Computer understanding of conventional metaphoric language. Cognitive Science: A Multidisciplinary Journal 16, Nr. 2: 233–270
- Martin, J.H. (1988). A computational theory of metaphor (Rep. No. UCB/CSD 88-465). Berkeley: University of California, Berkeley, Computer Science Department.