Strawman Model of Reference

Hinrich Schütze

Center for Information and Language Processing, LMU Munich

2015-09-14

Human NLP: Naive & highly simplistic model

Human NLP: Naive & highly simplistic model

- NLP is divided into two steps.

Human NLP: Naive & highly simplistic model

- NLP is divided into two steps.
- Step 1:
  linking bits & pieces of the input to bits & pieces in memory

Human NLP: Naive & highly simplistic model

- NLP is divided into two steps.
- Step 1:
  linking bits & pieces of the input to bits & pieces in memory
- Step 2:
  all the complicated stuff:
  disambiguation, parsing, inference

Human NLP: Naive & highly simplistic model

- NLP is divided into two steps.
- Step 1:
  linking bits & pieces of the input to bits & pieces in memory
- Step 2:
  all the complicated stuff:
  disambiguation, parsing, inference

Human NLP: Naive & highly simplistic model, Step 1

Human NLP: Naive & highly simplistic model, Step 1

- Memory: working, semantic, episodic

Human NLP: Naive & highly simplistic model, Step 1

- Memory: working, semantic, episodic
- Working memory: pronouns etc.

Human NLP: Naive & highly simplistic model, Step 1

- Memory: working, semantic, episodic
- Working memory: pronouns etc.
- Semantic memory: words etc.

Human NLP: Naive & highly simplistic model, Step 1

- Memory: working, semantic, episodic
- Working memory: pronouns etc.
- Semantic memory: words etc.
- Episodic memory: entities etc.

Human NLP: Naive & highly simplistic model, Step 1

- Memory: working, semantic, episodic
- Working memory: pronouns etc.
- Semantic memory: words etc.
- Episodic memory: entities etc.
- "She enjoyed the Olympics."

Human NLP: Naive & highly simplistic model, Step 1

- Memory: working, semantic, episodic
- Working memory: pronouns etc.
- Semantic memory: words etc.
- Episodic memory: entities etc.
- "She enjoyed the Olympics."
- "She": working memory lookup

Human NLP: Naive & highly simplistic model, Step 1

- Memory: working, semantic, episodic
- Working memory: pronouns etc.
- Semantic memory: words etc.
- Episodic memory: entities etc.
- “She enjoyed the Olympics.”
- “She”: working memory lookup
- “enjoyed”: semantic memory lookup

Human NLP: Naive & highly simplistic model, Step 1

- Memory: working, semantic, episodic
- Working memory: pronouns etc.
- Semantic memory: words etc.
- Episodic memory: entities etc.
- "She enjoyed the Olympics."
- "She": working memory lookup
- "enjoyed": semantic memory lookup
- "the Olympics": episodic memory lookup

Machine NLP: Naive & highly simplistic model

- NLP is divided into two steps.
- Step 1:
  linking bits & pieces of the input to bits & pieces in memory
- Step 2:
  all the complicated stuff:
  disambiguation, parsing, inference

Machine NLP: Naive & highly simplistic model, Step 1

- Memory: working, semantic, episodic
- Working: pronouns etc.
- Semantic: words etc.
- Episodic: entities etc.
- "She enjoyed the Olympics."
- "She": working memory lookup
- "enjoyed": semantic memory lookup
- "the Olympics": episodic memory lookup

Why I like embeddings

Why I like embeddings

- (Actually: why I like representation learning for words in general, including LSI, Brown classes etc.)

Why I like embeddings

- (Actually: why I like representation learning for words in general, including LSI, Brown classes etc.)
- Embeddings are
  a first stab at approximating semantic memory.

Why I like embeddings

- (Actually: why I like representation learning for words in general, including LSI, Brown classes etc.)
- Embeddings are
  a first stab at approximating semantic memory.
- Standard NLP: Before training we know nothing about a word.

Why I like embeddings

- (Actually: why I like representation learning for words in general, including LSI, Brown classes etc.)
- Embeddings are
  a first stab at approximating semantic memory.
- Standard NLP: Before training we know nothing about a word.
- (But we have a very richly annotated corpus like WSJ.)

Why I like embeddings

- (Actually: why I like representation learning for words in general, including LSI, Brown classes etc.)
- Embeddings are
  a first stab at approximating semantic memory.
- Standard NLP: Before training we know nothing about a word.
- (But we have a very richly annotated corpus like WSJ.)
- Embedding-based NLP:
  when you encounter a word,
  you look it up in semantic memory
  and you know a lot about the word right away.

Why I like embeddings

- (Actually: why I like representation learning for words in general, including LSI, Brown classes etc.)
- Embeddings are
  a first stab at approximating semantic memory.
- Standard NLP: Before training we know nothing about a word.
- (But we have a very richly annotated corpus like WSJ.)
- Embedding-based NLP:
  when you encounter a word,
  you look it up in semantic memory
  and you know a lot about the word right away.
- Working/Episodic memory:
  extension of "semantic" memory/embeddings?

Operations: reference / lookup

- Operation R:T→W
  Establishes Reference from Text to Working memory

Operations: reference / lookup

- Operation R:T$\rightarrow$W
  Establishes Reference from Text to Working memory
- text span $\rightarrow$ working memory slot

Operations: reference / lookup

- Operation R:T$\rightarrow$W
  Establishes Reference from Text to Working memory
- text span $\rightarrow$ working memory slot
- Example: "he" $\rightarrow$ working memory slot w_2

Operations: reference / lookup

- Operation R:T$\rightarrow$E
  Establishes Reference from Text to Episodic memory

Operations: reference / lookup

- Operation R:T$\rightarrow$E
  Establishes Reference from Text to Episodic memory
- text span $\rightarrow$ episodic memory slot

Operations: reference / lookup

- Operation R:T$\rightarrow$E
  Establishes Reference from Text to Episodic memory
- text span $\rightarrow$ episodic memory slot
- Example: “Olympics”
  $\rightarrow$ episodic memory slot “Beijing Olympics”

Operations: reference / lookup

- Operation R:T$\rightarrow$S
  Establishes "Reference" from Text to Semantic memory

Operations: reference / lookup

- Operation R:T$\rightarrow$S
  Establishes “Reference” from Text to Semantic memory
- text span $\rightarrow$ semantic memory slot

Operations: reference / lookup

- Operation R:T$\rightarrow$S
  Establishes "Reference" from Text to Semantic memory
- text span $\rightarrow$ semantic memory slot
- "red tape" $\rightarrow$ semantic memory slot for "red tape"

Operations: reference / lookup

- Operation R:T$\rightarrow$S
  Establishes "Reference" from Text to Semantic memory
- text span $\rightarrow$ semantic memory slot
- "red tape" $\rightarrow$ semantic memory slot for "red tape"
- So: semantic memory lookup = recognition of word

Operations: reference / lookup

- Operation R:T$\rightarrow$S
  Establishes "Reference" from Text to Semantic memory
- text span $\rightarrow$ semantic memory slot
- "red tape" $\rightarrow$ semantic memory slot for "red tape"
- So: semantic memory lookup = recognition of word
- May or may not be a good idea

Operations: storage / manipulation

- Operation I:W.
  Initialize Working memory slot w_i

Operations: storage / manipulation

- Operation I:W.
  Initialize Working memory slot w_i
- Peter arrived today.

Operations: storage / manipulation

- Operation I:W.
  Initialize Working memory slot w_i
- Peter arrived today.
- Claim one working memory slot for "Peter" and put "Peter" in it.

Operations: storage / manipulation

- Operation M:W$\leftrightarrow$E.
  Establishes Memory link Working M. $\leftrightarrow$ Episodic M.

Operations: storage / manipulation

- Operation M:W$\leftrightarrow$E.
  Establishes Memory link Working M. $\leftrightarrow$ Episodic M.
- working memory slot $w_2 \leftrightarrow$ episodic memory slot

Operations: storage / manipulation

- Operation M:W$\leftrightarrow$E.
  Establishes Memory link Working M. $\leftrightarrow$ Episodic M.
- working memory slot $w_2 \leftrightarrow$ episodic memory slot
- Obama is US president. He will leave office in 2017.

Operations: storage / manipulation

- Operation M:W↔E.
  Establishes Memory link Working M. ↔ Episodic M.
- working memory slot $w_2 \leftrightarrow$ episodic memory slot
- Obama is US president. He will leave office in 2017.
- In this example:
  w_2 (he) $\leftrightarrow$ episodic memory slot "Obama"

Operations: storage / manipulation

- Operation C:T$\rightarrow$W.
  Copies content from Text to Working memory.

Operations: storage / manipulation

- Operation C:T→W.
  Copies content from Text to Working memory.
- "Peter arrived today."

Operations: storage / manipulation

- Operation C:T$\rightarrow$W.
  Copies content from Text to Working memory.
- “Peter arrived today.”
- “arrived”: copy to working memory

Operations: storage / manipulation

- Operation C:W$\rightarrow$E.
  Copies content from Working M. to Episodic M.

Operations: storage / manipulation

- Operation C:W→E.
  Copies content from Working M. to Episodic M.
- "The 1956 Olympics were held in Melbourne."

Operations: storage / manipulation

- Operation C:W$\rightarrow$E.
  Copies content from Working M. to Episodic M.
- "The 1956 Olympics were held in Melbourne."
- First put in Working memory, then copy to Episodic memory

Operations: Summary

R:T→W	Reference text→working
R:T→E	Reference text→epsodic
R:T→S	Reference text→semantic
I:W	Initialize working memory slot
M:W↔E	Link working↔episodic
C:T→W	Copy text→working
C:W→E	Copy working→episodic

Backup slides

Reference: Discrete or not discrete?

- Storage locations are discrete and in that sense against the spirit of distributed representations / deep learning.
- Problem 1: Non-discrete reference phenomena
- GIVE EXAMPLES

Words vs. entities

- Can we also use embeddings for entities?
- The answer is not clear to me: Pros and cons.
- However, arguments for episodic memory could also be arguments for treating words and entities differently.