Semantic Analysis of Image-Based Learner Sentences

Levi King
Indiana University

August 6, 2021

Background & Motivation

Background & Motivation

- Most intelligent computer-assisted language learning (ICALL) applications (*Rosetta Stone*, *Duo Lingo*, etc.) rely on outdated, ineffective methods:

Background & Motivation

- Most intelligent computer-assisted language learning (ICALL) applications (*Rosetta Stone*, *Duo Lingo*, etc.) rely on outdated, ineffective methods:
  - rote memorization & grammatical error detection; menu-based vs. free input;

Background & Motivation

- Most intelligent computer-assisted language learning (ICALL) applications (*Rosetta Stone*, *Duo Lingo*, etc.) rely on outdated, ineffective methods:
  - rote memorization & grammatical error detection; menu-based vs. free input;
  - *"engineering first"*: not informed by second language acquisition (SLA), pedagogy, psychometrics

Background & Motivation

- Most intelligent computer-assisted language learning (ICALL) applications (*Rosetta Stone*, *Duo Lingo*, etc.) rely on outdated, ineffective methods:
  - rote memorization & grammatical error detection; menu-based vs. free input;
  - *"engineering first"*: not informed by second language acquisition (SLA), pedagogy, psychometrics
- SLA research → communicative & task based learning

Background & Motivation

- Most intelligent computer-assisted language learning (ICALL) applications (*Rosetta Stone*, *Duo Lingo*, etc.) rely on outdated, ineffective methods:
  - rote memorization & grammatical error detection; menu-based vs. free input;
  - *"engineering first"*: not informed by second language acquisition (SLA), pedagogy, psychometrics
- SLA research → communicative & task based learning

How can we bridge this gap between ICALL and SLA researchers?

Background & Motivation

- Most intelligent computer-assisted language learning (ICALL) applications (*Rosetta Stone*, *Duo Lingo*, etc.) rely on outdated, ineffective methods:
  - rote memorization & grammatical error detection; menu-based vs. free input;
  - *"engineering first"*: not informed by second language acquisition (SLA), pedagogy, psychometrics
- SLA research $\rightarrow$ communicative & task based learning

How can we bridge this gap between ICALL and SLA researchers?

- My vision:

Background & Motivation

- Most intelligent computer-assisted language learning (ICALL) applications (*Rosetta Stone*, *Duo Lingo*, etc.) rely on outdated, ineffective methods:
  - rote memorization & grammatical error detection; menu-based vs. free input;
  - *"engineering first"*: not informed by second language acquisition (SLA), pedagogy, psychometrics
- SLA research → communicative & task based learning

How can we bridge this gap between ICALL and SLA researchers?

- My vision: open source app; transparent; pipeline of existing tools;

Background & Motivation

- Most intelligent computer-assisted language learning (ICALL) applications (*Rosetta Stone*, *Duo Lingo*, etc.) rely on outdated, ineffective methods:
  - rote memorization & grammatical error detection; menu-based vs. free input;
  - *"engineering first"*: not informed by second language acquisition (SLA), pedagogy, psychometrics
- SLA research → communicative & task based learning

How can we bridge this gap between ICALL and SLA researchers?

- My vision: open source app; transparent; pipeline of existing tools;
- teachers create new games/stories by adding visual prompts and crowdsourcing native speaker (NS) responses;

Background & Motivation

- Most intelligent computer-assisted language learning (ICALL) applications (*Rosetta Stone*, *Duo Lingo*, etc.) rely on outdated, ineffective methods:
  - rote memorization & grammatical error detection; menu-based vs. free input;
  - *"engineering first"*: not informed by second language acquisition (SLA), pedagogy, psychometrics
- SLA research → communicative & task based learning

How can we bridge this gap between ICALL and SLA researchers?

- My vision: open source app; transparent; pipeline of existing tools;
- teachers create new games/stories by adding visual prompts and crowdsourcing native speaker (NS) responses;
- trains NS model to evaluate non-native speaker (NNS) responses

Research Questions

Research Questions

Research Questions

RQ1. Are the responses of L2 English learners sufficiently similar to those of NSs to allow automatic evaluation based on a collection of NS responses? In other words, do learners demonstrate significant overlap with native-like usage in a picture description task (PDT) setting?

Research Questions

RQ1. Are the responses of L2 English learners sufficiently similar to those of NSs to allow automatic evaluation based on a collection of NS responses? In other words, do learners demonstrate significant overlap with native-like usage in a picture description task (PDT) setting?

RQ2. In the constrained communicative environment of a PDT, what are appropriate response and model representations for the purpose of providing meaning-oriented feedback or evaluation? In other words, which linguistic components are crucial and which are superfluous?

Research Questions

RQ1. Are the responses of L2 English learners sufficiently similar to those of NSs to allow automatic evaluation based on a collection of NS responses? In other words, do learners demonstrate significant overlap with native-like usage in a picture description task (PDT) setting?

RQ2. In the constrained communicative environment of a PDT, what are appropriate response and model representations for the purpose of providing meaning-oriented feedback or evaluation? In other words, which linguistic components are crucial and which are superfluous?

RQ3. What kinds of existing NLP tools and language resources can be integrated to form a content analysis system for open response language learning tasks?

Research Questions

Research Questions

RQ4. How do “bag-of-words” and “bag-of-dependencies” approaches compare in terms of performance? Is a bag-of-words approach alone adequate for our needs?

Research Questions

RQ4. How do "bag-of-words" and "bag-of-dependencies" approaches compare in terms of performance? Is a bag-of-words approach alone adequate for our needs?

RQ5. Can the accuracy of the system be improved by the inclusion of semantic information from tools like semantic role labelers, WordNet, or word and sentence embeddings?

Research Questions

RQ4. How do “bag-of-words” and “bag-of-dependencies” approaches compare in terms of performance? Is a bag-of-words approach alone adequate for our needs?

RQ5. Can the accuracy of the system be improved by the inclusion of semantic information from tools like semantic role labelers, WordNet, or word and sentence embeddings?

RQ6. What is the annotation scheme for this task and can the system perform within the range of human performance? Relatedly, what does it mean for a response to be *appropriate* and how can this be captured with annotation?

Pilot study: Data

Response (L1)
He is droning his wife pitcher. (Ar)
The artist is drawing a pretty women. (Ch)
The artist is painting a portrait of a lady. (En)
The painter is painting a woman's paint. (Sp)

Figure: Example item from the pilot study showing responses from native speakers of Arabic (Ar), Chinese (Ch), English (En) and Spanish (Sp).

Pilot study: Data

Response (L1)
He is droning his wife pitcher. (Ar)
The artist is drawing a pretty women. (Ch)
The artist is painting a portrait of a lady. (En)
The painter is painting a woman's paint. (Sp)

Figure: Example item from the pilot study showing responses from native speakers of Arabic (Ar), Chinese (Ch), English (En) and Spanish (Sp).

- 10 (transitive) PDT items $\times$ 53 participants $=$ 530 responses;

Pilot study: Data

Response (L1)
He is droning his wife pitcher. (Ar)
The artist is drawing a pretty women. (Ch)
The artist is painting a portrait of a lady. (En)
The painter is painting a woman's paint. (Sp)

Figure: Example item from the pilot study showing responses from native speakers of Arabic (Ar), Chinese (Ch), English (En) and Spanish (Sp).

- 10 (transitive) PDT items $\times$ 53 participants $=$ 530 responses;
  - 14 NSs (grad students), 39 NNSs (ESL students);

Pilot study: Data

Response (L1)
He is droning his wife pitcher. (Ar)
The artist is drawing a pretty women. (Ch)
The artist is painting a portrait of a lady. (En)
The painter is painting a woman's paint. (Sp)

Figure: Example item from the pilot study showing responses from native speakers of Arabic (Ar), Chinese (Ch), English (En) and Spanish (Sp).

- 10 (transitive) PDT items $\times$ 53 participants $=$ 530 responses;
  - 14 NSs (grad students), 39 NNSs (ESL students);
- Annotation: *Given the prompt, would the response be acceptable to most English speakers? Acceptable/unacceptable*
  - 1 annotator (me)

Pilot study: Data

Response (L1)
He is droning his wife pitcher. (Ar)
The artist is drawing a pretty women. (Ch)
The artist is painting a portrait of a lady. (En)
The painter is painting a woman's paint. (Sp)

Figure: Example item from the pilot study showing responses from native speakers of Arabic (Ar), Chinese (Ch), English (En) and Spanish (Sp).

- 10 (transitive) PDT items $\times$ 53 participants $=$ 530 responses;
  - 14 NSs (grad students), 39 NNSs (ESL students);
- Annotation: *Given the prompt, would the response be acceptable to most English speakers? Acceptable/unacceptable*
  - 1 annotator (me)

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ *V(S,O)* extraction rules;

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list $\rightarrow$ covered / not covered;

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list $\rightarrow$ covered / not covered;

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list $\rightarrow$ covered / not covered;

- Dependency-based

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list $\rightarrow$ covered / not covered;

- Dependency-based
  - Captures aspects of form and meaning;

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list $\rightarrow$ covered / not covered;

- Dependency-based
  - Captures aspects of form and meaning;
  - Subjects, objects, verbs clearly labeled;

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser → lemmatizer → *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list → covered / not covered;

- Dependency-based
  - Captures aspects of form and meaning;
  - Subjects, objects, verbs clearly labeled;
- V(S,O) extraction

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser → lemmatizer → *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list → covered / not covered;

- Dependency-based
  - Captures aspects of form and meaning;
  - Subjects, objects, verbs clearly labeled;
- V(S,O) extraction
  - Decision tree based on dependency indexing & labels, POS;

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser → lemmatizer → *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list → covered / not covered;

- Dependency-based
  - Captures aspects of form and meaning;
  - Subjects, objects, verbs clearly labeled;
- V(S,O) extraction
  - Decision tree based on dependency indexing & labels, POS;
  - Custom for my transitive PDT, not generalizable, not robust;

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser → lemmatizer → *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list → covered / not covered;

- Dependency-based
  - Captures aspects of form and meaning;
  - Subjects, objects, verbs clearly labeled;
- V(S,O) extraction
  - Decision tree based on dependency indexing & labels, POS;
  - Custom for my transitive PDT, not generalizable, not robust;
  - ≈92% accurate, ≈8% extraction errors;

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser → lemmatizer → *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list → covered / not covered;

- Dependency-based
  - Captures aspects of form and meaning;
  - Subjects, objects, verbs clearly labeled;
- V(S,O) extraction
  - Decision tree based on dependency indexing & labels, POS;
  - Custom for my transitive PDT, not generalizable, not robust;
  - ≈92% accurate, ≈8% extraction errors;
- Overall accuracy: 58.9%

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser → lemmatizer → *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list → covered / not covered;

- Dependency-based
  - Captures aspects of form and meaning;
  - Subjects, objects, verbs clearly labeled;
- V(S,O) extraction
  - Decision tree based on dependency indexing & labels, POS;
  - Custom for my transitive PDT, not generalizable, not robust;
  - ≈92% accurate, ≈8% extraction errors;
- Overall accuracy: 58.9%

Pilot study: Processing

First approach: **Rule-based** triple extraction and matching

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ *V(S,O)* extraction rules;

Compare NNS *V(S,O)* & NS *V(S,O)* list $\rightarrow$ covered / not covered;

- Dependency-based
  - Captures aspects of form and meaning;
  - Subjects, objects, verbs clearly labeled;
- V(S,O) extraction
  - Decision tree based on dependency indexing & labels, POS;
  - Custom for my transitive PDT, not generalizable, not robust;
  - ≈92% accurate, ≈8% extraction errors;
- Overall accuracy: 58.9%
  - I.e., *Acceptable* covered, *unacceptable* not covered;

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser → lemmatizer → term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser → lemmatizer → term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ term frequency-inverse document frequency (tf-idf; "term" $=$ lemmatized dependency);

NNS response score $=$ cosine distance of NS and NNS tf-idf scores;

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser → lemmatizer → term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

- tf-idf: Score dependencies according to importance;

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

- tf-idf: Score dependencies according to importance;
- Vectorize & Score

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

- tf-idf: Score dependencies according to importance;
- Vectorize & Score
  - Get *sorted union set* of NS and NNS dependencies;

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

- tf-idf: Score dependencies according to importance;
- Vectorize & Score
  - Get *sorted union set* of NS and NNS dependencies;
  - NNS vector: Replace deps with their **NNS** tf-idf scores;

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser → lemmatizer → term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

- tf-idf: Score dependencies according to importance;
- Vectorize & Score
  - Get *sorted union set* of NS and NNS dependencies;
  - NNS vector: Replace deps with their **NNS** tf-idf scores;
  - NS vector: Replace deps with their **NS** tf-idf scores;

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser $\rightarrow$ lemmatizer $\rightarrow$ term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

- tf-idf: Score dependencies according to importance;
- Vectorize & Score
  - Get *sorted union set* of NS and NNS dependencies;
  - NNS vector: Replace deps with their **NNS** tf-idf scores;
  - NS vector: Replace deps with their **NS** tf-idf scores;
  - Response score = *cosine distance* for NNS & NS vectors;

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser → lemmatizer → term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

- tf-idf: Score dependencies according to importance;
- Vectorize & Score
  - Get *sorted union set* of NS and NNS dependencies;
  - NNS vector: Replace deps with their **NNS** tf-idf scores;
  - NS vector: Replace deps with their **NS** tf-idf scores;
  - Response score = *cosine distance* for NNS & NS vectors;
- Rank by scores & calculate Mean Average Precision (MAP);

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser → lemmatizer → term frequency-inverse document frequency (tf-idf; “term” = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

- tf-idf: Score dependencies according to importance;
- Vectorize & Score
  - Get *sorted union set* of NS and NNS dependencies;
  - NNS vector: Replace deps with their **NNS** tf-idf scores;
  - NS vector: Replace deps with their **NS** tf-idf scores;
  - Response score = *cosine distance* for NNS & NS vectors;
- Rank by scores & calculate Mean Average Precision (MAP);
  - MAP *acceptable* responses: ≈51%

Pilot study: Processing

Second approach: **Semantic similarity** scoring

Dependency parser → lemmatizer → term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

- tf-idf: Score dependencies according to importance;
- Vectorize & Score
  - Get *sorted union set* of NS and NNS dependencies;
  - NNS vector: Replace deps with their **NNS** tf-idf scores;
  - NS vector: Replace deps with their **NS** tf-idf scores;
  - Response score = *cosine distance* for NNS & NS vectors;
- Rank by scores & calculate Mean Average Precision (MAP);
  - MAP *acceptable* responses: ≈51%
- Process is more robust & generalizable;

Pilot study: Processing

Second approach: **Semantic similarity** scoring

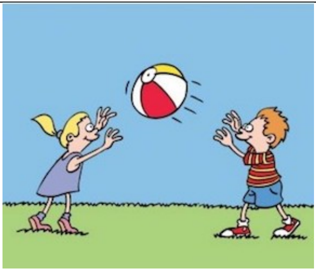
Dependency parser → lemmatizer → term frequency-inverse document frequency (tf-idf; "term" = lemmatized dependency);

NNS response score = cosine distance of NS and NNS tf-idf scores;

- tf-idf: Score dependencies according to importance;
- Vectorize & Score
  - Get *sorted union set* of NS and NNS dependencies;
  - NNS vector: Replace deps with their **NNS** tf-idf scores;
  - NS vector: Replace deps with their **NS** tf-idf scores;
  - Response score = *cosine distance* for NNS & NS vectors;
- Rank by scores & calculate Mean Average Precision (MAP);
  - MAP *acceptable* responses: ≈51%
- Process is more robust & generalizable;
- Dataset (especially NS models) and annotation are weak;

Main study: Data collection

PDT with very simple images only:

	
Original image	Simplified PDT image

Intended to focus participants' attention on the main action

Main study: Data collection

Two PDT prompt versions:

Targeted	**Untargeted**
	
*What is **the baby** doing?*	*What is happening?*

Intended for exploring the specificity needed for my approach

Main study: Data collection

The pilot study *rake* problem:

- For one PDT item, 100% of NS used the verb *rake*;
  - NNS responses without *rake* are penalized;

NNS Responses
The gardener is *cleaning* the street.
a man *removing* the tree leafs.
The man is *sweeping* the floor.
A man is *gathering* lots of leafs.

- I address this by asking NSs for two non-identical responses.

Main study: Data collection

3 verb types:

10 **intransitive** items	10 **transitive** items	10 **ditransitive** items
		
What is the girl doing?	What is the boy doing?	What is the girl doing?

Intended for exploring whether my approach can generalize to a broader range of sentence types

Main study: Data collection

499 participants, 13,533 responses:

Main study: Data collection

499 participants, 13,533 responses:

Main study: Data collection

499 participants, 13,533 responses:

- 141 NNSs (English Language Improvement Program at IU);

Main study: Data collection

499 participants, 13,533 responses:

- 141 NNSs (English Language Improvement Program at IU);
  - 125 Mandarin, 4 Korean, 3 Burmese, 2 Hindi; 1 each: Arabic, Indonesian, German, Gujarati, Spanish, Thai, Vietnamese;

Main study: Data collection

499 participants, 13,533 responses:

- 141 NNSs (English Language Improvement Program at IU);
  - 125 Mandarin, 4 Korean, 3 Burmese, 2 Hindi; 1 each: Arabic, Indonesian, German, Gujarati, Spanish, Thai, Vietnamese;
- 358 NSs:

Main study: Data collection

499 participants, 13,533 responses:

- 141 NNSs (English Language Improvement Program at IU);
  - 125 Mandarin, 4 Korean, 3 Burmese, 2 Hindi; 1 each: Arabic, Indonesian, German, Gujarati, Spanish, Thai, Vietnamese;
- 358 NSs:
  - 329 `crowdsourced`, purchased via SurveyMonkey;

Main study: Data collection

499 participants, 13,533 responses:

- 141 NNSs (English Language Improvement Program at IU);
  - 125 Mandarin, 4 Korean, 3 Burmese, 2 Hindi; 1 each: Arabic, Indonesian, German, Gujarati, Spanish, Thai, Vietnamese;
- 358 NSs:
  - 329 `crowdsourced`, purchased via SurveyMonkey;
  - 29 `familiar`, unpaid colleagues;

	Response Counts		
Group	First	Second	Total
NNS	4290	0	4290
NS (all)	4634	4609	9243
`Familiar`	642	641	1283
`Crowdsrc`	3992	3968	7960
Total	8924	4609	**13,533**

Annotation features

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

- Core event:

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

- CORE EVENT: Does response capture main action?

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

- CORE EVENT: Does response capture main action?
- ANSWERHOOD:

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

- Core event: Does response capture main action?
- Answerhood: Does response directly answer prompt?

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

- Core event: Does response capture main action?
- Answerhood: Does response directly answer prompt?
- Grammaticality:

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

- CORE EVENT: Does response capture main action?
- ANSWERHOOD: Does response directly answer prompt?
- GRAMMATICALITY: Is response free from grammar problems?

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

- CORE EVENT: Does response capture main action?
- ANSWERHOOD: Does response directly answer prompt?
- GRAMMATICALITY: Is response free from grammar problems?
- INTERPRETABILITY:

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

- Core event: Does response capture main action?
- Answerhood: Does response directly answer prompt?
- Grammaticality: Is response free from grammar problems?
- Interpretability: Does response evoke a clear mental image?

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: $+$A, $+$NL $>$ **1**: $+$A, $-$NL $>$ **0**: $-$A, $-$NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

- CORE EVENT: Does response capture main action?
- ANSWERHOOD: Does response directly answer prompt?
- GRAMMATICALITY: Is response free from grammar problems?
- INTERPRETABILITY: Does response evoke a clear mental image?
- VERIFIABILITY:

Annotation features

First iteration: **accuracy (A)** & **native-likeness (NL)**

- **2**: +A, +NL > **1**: +A, −NL > **0**: −A, −NL
- Not operationalizable: e.g., response is accurate w.r.t. prompt but adds unverifiable details; is this still *accurate*?
- Not *reliable*, not *valid*;

Several iterations later, 5 binary features:

- CORE EVENT: Does response capture main action?
- ANSWERHOOD: Does response directly answer prompt?
- GRAMMATICALITY: Is response free from grammar problems?
- INTERPRETABILITY: Does response evoke a clear mental image?
- VERIFIABILITY: Is all response info supported by image?

Annotation features

What is the boy doing?	C	A	G	I	V
He is eating food.	0	1	1	1	1
he eating pizza.	1	1	0	1	1
The boy is smiling pizza.	0	1	0	0	0
He may get fat eating.	0	0	1	1	0

Table: Annotated for five features: Core event (C), Answerhood (A), Grammaticality (G), Interpretability (I) and Verifiability (V).

Annotation features

Annotation features

Inter-rater reliability for two annotators and 10% of the dataset:

Annotation features

Inter-rater reliability for two annotators and 10% of the dataset: *yes* annotations for Annotator 1 (note skewedness), expected chance agreement (*Chance*), actual observed agreement (*Observed*) and Cohen's kappa (*Kappa*)

Set	A1Yes	Chance	Observed	Kappa
Core Event	0.733	0.601	0.923	0.808
Answerhood	0.834	0.721	0.982	0.936
Grammaticality	0.861	0.768	0.960	0.827
Interpretability	0.818	0.682	0.919	0.744
Verifiability	0.845	0.719	0.968	0.884
Intransitive	0.863	0.758	0.978	0.910
Transitive	0.780	0.653	0.949	0.853
Ditransitive	0.812	0.678	0.924	0.764

Weighting features

Problem: My system provides a similarity score between 0 and 1.
How can I evaluate system performance?

Weighting features

Problem: My system provides a similarity score between 0 and 1.
How can I evaluate system performance?

Feature-level performance:

Weighting features

Problem: My system provides a similarity score between 0 and 1.
How can I evaluate system performance?

Feature-level performance:

- **MAP** to see how system rankings predict **individual features**;

Weighting features

Problem: My system provides a similarity score between 0 and 1.
How can I evaluate system performance?

Feature-level performance:

- **MAP** to see how system rankings predict **individual features**;
  - Compare with? Some *holistic benchmark ranking* MAP;

Weighting features

Problem: My system provides a similarity score between 0 and 1. How can I evaluate system performance?

Feature-level performance:

- **MAP** to see how system rankings predict **individual features**;
  - Compare with? Some *holistic benchmark ranking* MAP;

Holistic performance (response quality):

Weighting features

Problem: My system provides a similarity score between 0 and 1. How can I evaluate system performance?

Feature-level performance:

- **MAP** to see how system rankings predict **individual features**;
  - Compare with? Some *holistic benchmark ranking* MAP;

Holistic performance (response quality):

- **Spearman** rank correlation: Compare system rankings with some *holistic benchmark ranking*;

Weighting features

Problem: My system provides a similarity score between 0 and 1. How can I evaluate system performance?

Feature-level performance:

- **MAP** to see how system rankings predict **individual features**;
  - Compare with? Some *holistic benchmark ranking* MAP;

Holistic performance (response quality):

- **Spearman** rank correlation: Compare system rankings with some *holistic benchmark ranking*;

I need **benchmark response rankings** for the NNS test set.

Weighting features

Problem: My system provides a similarity score between 0 and 1. How can I evaluate system performance?

Feature-level performance:

- **MAP** to see how system rankings predict **individual features**;
  - Compare with? Some *holistic benchmark ranking* MAP;

Holistic performance (response quality):

- **Spearman** rank correlation: Compare system rankings with some *holistic benchmark ranking*;

I need **benchmark response rankings** for the NNS test set.

Solution: Determine feature weights and apply to annotations to obtain benchmark holistic scores and then rankings.

Weighting features

Raters perform holistic preference test (blind to annotations)

Weighting features

Raters perform holistic preference test (blind to annotations)

What is the boy doing?	Pref?	Core	Ansr	Gram	Intrp	Verif
He is eating food.	**yes**	**0**	**1**	**1**	**1**	**1**
He may get fat eating.	no	0	0	1	1	0

Weighting features

Raters perform holistic preference test (blind to annotations)

What is the boy doing?	Pref?	Core	Ansr	Gram	Intrp	Verif
He is eating food.	**yes**	**0**	**1**	**1**	**1**	**1**
He may get fat eating.	no	0	0	1	1	0
He is hungry.	no	0	0	1	0	1
the boy is eating pizza	**yes**	**1**	**1**	**1**	**1**	**1**

Weighting features

Raters perform holistic preference test (blind to annotations)

What is the boy doing?	Pref?	Core	Ansr	Gram	Intrp	Verif
He is eating food.	**yes**	**0**	**1**	**1**	**1**	**1**
He may get fat eating.	no	0	0	1	1	0
He is hungry.	no	0	0	1	0	1
the boy is eating pizza	**yes**	**1**	**1**	**1**	**1**	**1**
The child is about to eat pizza.	**yes**	**1**	**0**	**1**	**1**	**1**
he eating.	no	0	1	0	1	1

Weighting features

Raters perform holistic preference test (blind to annotations)

What is the boy doing?	Pref?	Core	Ansr	Gram	Intrp	Verif
He is eating food.	**yes**	**0**	**1**	**1**	**1**	**1**
He may get fat eating.	no	0	0	1	1	0
He is hungry.	no	0	0	1	0	1
the boy is eating pizza	**yes**	**1**	**1**	**1**	**1**	**1**
The child is about to eat pizza.	**yes**	**1**	**0**	**1**	**1**	**1**
he eating.	no	0	1	0	1	1
Totals preferred responses		2	2	3	3	3
Totals dispreferred responses		0	1	2	2	2
Net preferred (pref - dispref)		2	1	1	1	1
Feature weight		.333	.167	.167	.167	.167

Weighting features

Raters perform holistic preference test (blind to annotations)

What is the boy doing?	Pref?	Core	Ansr	Gram	Intrp	Verif
He is eating food.	**yes**	**0**	**1**	**1**	**1**	**1**
He may get fat eating.	no	0	0	1	1	0
He is hungry.	no	0	0	1	0	1
the boy is eating pizza	**yes**	**1**	**1**	**1**	**1**	**1**
The child is about to eat pizza.	**yes**	**1**	**0**	**1**	**1**	**1**
he eating.	no	0	1	0	1	1
Totals preferred responses		2	2	3	3	3
Totals dispreferred responses		0	1	2	2	2
Net preferred (pref - dispref)		2	1	1	1	1
Feature weight		.333	.167	.167	.167	.167
*Real feature weight		.365	.093	.055	.224	.263

Weighting features

Preferences are reliable:

Agreement for two annotators on a sample of 300 response pairs:

Chance Agree	Observed Agree	Cohen's Kappa
0.621	0.883 (265/300)	0.692

Benchmark rankings

Benchmark rankings

- Obtain weighted annotation scores (WAS) by applying feature weights to binary annotations;

Benchmark rankings

- Obtain weighted annotation scores (WAS) by applying feature weights to binary annotations;
- Rank NNS test set responses by WAS to obtain weighted annotation ranking (WAR);

Benchmark rankings

- Obtain weighted annotation scores (WAS) by applying feature weights to binary annotations;
- Rank NNS test set responses by WAS to obtain weighted annotation ranking (WAR);

What is happening?	C	A	G	I	V	WAS	WAR
The boy is eating pizza	0.365	0.093	0.055	0.224	0.263	1.000	1
Child is eating pizza	0.365	0.093	0.000	0.224	0.263	0.945	2
Tommy is eating pizza	0.365	0.093	0.055	0.224	0.000	0.737	3
The boy's eating his favorite food	0.000	0.093	0.055	0.000	0.000	0.513	4
Pizza is this boy's favorite food	0.000	0.000	0.055	0.000	0.000	0.055	5

Benchmark rankings

- Obtain weighted annotation scores (WAS) by applying feature weights to binary annotations;
- Rank NNS test set responses by WAS to obtain weighted annotation ranking (WAR);

What is happening?	C	A	G	I	V	WAS	WAR
The boy is eating pizza	0.365	0.093	0.055	0.224	0.263	1.000	1
Child is eating pizza	0.365	0.093	0.000	0.224	0.263	0.945	2
Tommy is eating pizza	0.365	0.093	0.055	0.224	0.000	0.737	3
The boy's eating his favorite food	0.000	0.093	0.055	0.000	0.000	0.513	4
Pizza is this boy's favorite food	0.000	0.000	0.055	0.000	0.000	0.055	5

- Use WAR as benchmark;
  - Features: Get MAP for WAR, compare against system MAP;

Benchmark rankings

- Obtain weighted annotation scores (WAS) by applying feature weights to binary annotations;
- Rank NNS test set responses by WAS to obtain weighted annotation ranking (WAR);

What is happening?	C	A	G	I	V	WAS	WAR
The boy is eating pizza	0.365	0.093	0.055	0.224	0.263	1.000	1
Child is eating pizza	0.365	0.093	0.000	0.224	0.263	0.945	2
Tommy is eating pizza	0.365	0.093	0.055	0.224	0.000	0.737	3
The boy's eating his favorite food	0.000	0.093	0.055	0.000	0.000	0.513	4
Pizza is this boy's favorite food	0.000	0.000	0.055	0.000	0.000	0.055	5

- Use WAR as benchmark;
  - Features: Get MAP for WAR, compare against system MAP;
  - Holistic: Get Spearman for system ranking vs. WAR;

SBERT as benchmark

I also use SBERT as a kind of “benchmark” for comparing my system’s performance.

SBERT as benchmark

I also use SBERT as a kind of "benchmark" for comparing my system's performance.

SBERT as benchmark

I also use SBERT as a kind of “benchmark” for comparing my system’s performance.

- State-of-the-art sentence embeddings for semantic textual similarity.

SBERT as benchmark

I also use SBERT as a kind of “benchmark” for comparing my system’s performance.

- State-of-the-art sentence embeddings for semantic textual similarity.
- Replaces dependency parser + lemmatizer + tf-idf cosine pipeline.

SBERT as benchmark

I also use SBERT as a kind of "benchmark" for comparing my system's performance.

- State-of-the-art sentence embeddings for semantic textual similarity.
- Replaces dependency parser + lemmatizer + tf-idf cosine pipeline.
- Provides distance between NNS response and NS model.

SBERT as benchmark

I also use SBERT as a kind of “benchmark” for comparing my system’s performance.

- State-of-the-art sentence embeddings for semantic textual similarity.
- Replaces dependency parser + lemmatizer + tf-idf cosine pipeline.
- Provides distance between NNS response and NS model.
- Operates directly on text, not dependencies.

SBERT as benchmark

I also use SBERT as a kind of "benchmark" for comparing my system's performance.

- State-of-the-art sentence embeddings for semantic textual similarity.
- Replaces dependency parser + lemmatizer + tf-idf cosine pipeline.
- Provides distance between NNS response and NS model.
- Operates directly on text, not dependencies.
  - Term representation parameter (`ldh`, `xdh`, `xdx`) does not apply.

System configuration

Optimizing my system means searching for the settings that yield the best performance; i.e., system output best approximates benchmark (WAR).

System configuration

Optimizing my system means searching for the settings that yield the best performance; i.e., system output best approximates benchmark (WAR).

- **Transitivity**: `intransitive`, `transitive`, `ditransitive`;

System configuration

Optimizing my system means searching for the settings that yield the best performance; i.e., system output best approximates benchmark (WAR).

- **Transitivity**: `intransitive`, `transitive`, `ditransitive`;
- **Targeting**:
  - `targeted`: *What is <the subject> doing?*
  - `untargeted`: *What is happening?*

System configuration

Optimizing my system means searching for the settings that yield the best performance; i.e., system output best approximates benchmark (WAR).

- **Transitivity**: `intransitive`, `transitive`, `ditransitive`;
- **Targeting**:
  - `targeted`: *What is <the subject> doing?*
  - `untargeted`: *What is happening?*
- **Familiarity**: `familiar`, `crowdsourced`

System configuration

Optimizing my system means searching for the settings that yield the best performance; i.e., system output best approximates benchmark (WAR).

- **Transitivity**: `intransitive`, `transitive`, `ditransitive`;
- **Targeting**:
  - `targeted`: *What is <the subject> doing?*
  - `untargeted`: *What is happening?*
- **Familiarity**: `familiar`, `crowdsourced`
- **Primacy**:
  - `primary`: NS model contains only 1st responses;
  - `mixed`: NS model: 1st & 2nd responses (50-50);

System configuration

Optimizing my system means searching for the settings that yield the best performance; i.e., system output best approximates benchmark (WAR).

- **Transitivity**: `intransitive`, `transitive`, `ditransitive`;
- **Targeting**:
  - `targeted`: *What is <the subject> doing?*
  - `untargeted`: *What is happening?*
- **Familiarity**: `familiar`, `crowdsourced`
- **Primacy**:
  - `primary`: NS model contains only 1st responses;
  - `mixed`: NS model: 1st & 2nd responses (50-50);
- **Term Representation**:
  - `ldh`: label-dependent-head; i.e., labeled dependencies;
  - `xdh`: dependent-head; i.e., unlabeled dependencies;
  - `xdx`: dependent only; cf. *bag of words*;

System configuration

System configuration

All parameters or variables and their settings:

Transitivity	**Targeting**	**Familiarity**	**Primacy**	**Term Rep.**
`intransitive`	`targeted`	`familar`	`primary`	`ldh`
`transitive`	`untargeted`	`crowdsourced`	`mixed`	`xdh`
`ditransitive`				`xdx`

A **system configuration** combines one setting from each column.

If particular settings correlate highly with item characteristics (intransitive / transitive / ditransitive; response complexity), I can optimize the system for new items.

Sampling data

Sampling data

NNS test sets:

- All experiments rank the same randomly sampled NNS test sets;
- 70 responses per PDT item (max available for NNS data);
  - 70 `targeted`, 70 `untargeted`

Sampling data

NNS test sets:

- All experiments rank the same randomly sampled NNS test sets;
- 70 responses per PDT item (max available for NNS data);
  - 70 `targeted`, 70 `untargeted`

NS models:

- 14-response models (max available for `familiar` data);
  - I.e., I compare 14-response `familiar` models and 14-response `crowdsourced` models;
- 50-response models (max available for `crowdsourced` data);

Sampling data: Complexity

Standardized type-to-token ratio (STTR) for the response samples. Tokens here are *dependencies*.

	n14		n50	n70
	Fam	Crd	Crd	NNS
Intrans	0.55	0.52	0.53	0.39
Trans	0.56	0.58	0.58	0.51
Ditrans	0.59	0.64	0.63	0.60
Target	0.54	0.53	0.54	0.48
Untarg	0.61	0.63	0.62	0.52
primary	N/A	0.51	0.52	0.50
mixed	0.57	0.65	0.64	N/A
xdx	0.36	0.42	0.42	0.36
xdh	0.65	0.66	0.66	0.57
ldh	0.66	0.66	0.67	0.57
Total	0.57	0.58	0.58	0.50

Annotation features experiments: Core event MAP

	Crowd NS model = 14					Crowd NS model = 50				
	ldh	xdh	xdx	WAR	SBERT	ldh	xdh	xdx	WAR	SBERT
Intr	***0.85***	0.85	0.85	0.86	0.83	**0.85**	0.85	0.85	0.86	0.83
Tran	***0.73***	0.73	0.72	0.74	0.70	**0.73**	0.73	0.72	0.74	0.70
Ditr	***0.66***	0.66	0.66	0.66	0.63	0.65	0.65	**0.66**	0.66	0.62
Targ	***0.73***	0.73	0.73	0.73	0.70	**0.73**	0.73	0.72	0.73	0.70
Untg	***0.76***	0.76	0.76	0.77	0.74	0.76	0.75	**0.76**	0.77	0.73
Prim	***0.75***	0.75	0.74	0.75	0.72	**0.75**	0.74	0.74	0.75	0.71
Mix	***0.75***	0.74	0.75	0.75	0.72	**0.74**	0.74	0.74	0.75	0.72
Total	***0.75***	0.75	0.74	0.75	0.72	**0.75**	0.74	0.74	0.75	0.72

- In all cases, ldh + 14NS is best;
- xdx becomes more competitive for larger model (50NS);
  - ditransitive, untargeted: responses to these are *least homogenous*—i.e., highest TTRs;
  - In general: ldh TTR > xdh TTR > xdx TTR

Annotation features experiments: CORE EVENT MAP

	Familiar NS model = 14					Crowd NS model = 14				
	ldh	xdh	xdx	WAR	SBERT	ldh	xdh	xdx	WAR	SBERT
Intr	0.85	0.85	***0.86***	0.86	0.83	**0.85**	0.85	0.84	0.86	0.83
Tran	***0.74***	0.73	0.72	0.74	0.70	**0.73**	0.73	0.72	0.74	0.70
Ditr	0.65	0.64	**0.66**	0.66	0.62	0.66	0.65	***0.67***	0.66	0.64
Targ	**0.73**	0.73	0.73	0.73	0.70	***0.73***	0.73	0.73	0.73	0.70
Untg	0.76	0.76	***0.76***	0.77	0.73	**0.76**	0.76	**0.76**	0.77	0.74
Total	0.75	0.74	**0.75**	0.75	0.72	***0.75***	0.74	0.75	0.75	0.72

- Totals: crowdsourced outperforms familiar (slightly);
- crowdsourced works best with ldh;
- familiar works best with xdx;

Annotation features experiments: MAP Results

Annotation features experiments: MAP Results

For all 5 features, my system outperforms SBERT.

Annotation features experiments: MAP Results

For all 5 features, my system outperforms SBERT.

ANSWERHOOD, in *all* cases:

Annotation features experiments: MAP Results

For all 5 features, my system outperforms SBERT.

ANSWERHOOD, in *all* cases:

- `xdx` > `xdh` > `ldh`;

Annotation features experiments: MAP Results

For all 5 features, my system outperforms SBERT.

ANSWERHOOD, in *all* cases:

- `xdx` $>$ `xdh` $>$ `ldh`;
- Model size makes no difference;

Annotation features experiments: MAP Results

For all 5 features, my system outperforms SBERT.

ANSWERHOOD, in *all* cases:

- `xdx` > `xdh` > `ldh`;
- Model size makes no difference;
- `familiar` > `crowdsourced`;

Annotation features experiments: MAP Results

For all 5 features, my system outperforms SBERT.

ANSWERHOOD, in *all* cases:

- `xdx` > `xdh` > `ldh`;
- Model size makes no difference;
- `familiar` > `crowdsourced`;

GRAMMATICALITY, in *most* cases:

Annotation features experiments: MAP Results

For all 5 features, my system outperforms SBERT.

ANSWERHOOD, in *all* cases:

- `xdx` > `xdh` > `ldh`;
- Model size makes no difference;
- `familiar` > `crowdsourced`;

GRAMMATICALITY, in *most* cases:

- `xdx` > `xdh` > `ldh`;

Annotation features experiments: MAP Results

For all 5 features, my system outperforms SBERT.

ANSWERHOOD, in *all* cases:

- `xdx` > `xdh` > `ldh`;
- Model size makes no difference;
- `familiar` > `crowdsourced`;

GRAMMATICALITY, in *most* cases:

- `xdx` > `xdh` > `ldh`;
- `familiar` 14NS > `crowd` 14NS > `crowd` 50NS;

Annotation features experiments: MAP Results

For all 5 features, my system outperforms SBERT.

ANSWERHOOD, in *all* cases:

- `xdx` > `xdh` > `ldh`;
- Model size makes no difference;
- `familiar` > `crowdsourced`;

GRAMMATICALITY, in *most* cases:

- `xdx` > `xdh` > `ldh`;
- `familiar` 14NS > `crowd` 14NS > `crowd` 50NS;

Predicting ANSWERHOOD or GRAMMATICALITY is relatively simple; requires only small model and bag-of-words representation.

Annotation features experiments: MAP Results

Annotation features experiments: MAP Results

INTERPRETABILITY:

Annotation features experiments: MAP Results

INTERPRETABILITY:

- 14NS `crowd` > 14NS `familiar` > 50NS `crowd`;

Annotation features experiments: MAP Results

INTERPRETABILITY:

- 14NS `crowd` > 14NS `familiar` > 50NS `crowd`;

VERIFIABILITY:

Annotation features experiments: MAP Results

INTERPRETABILITY:

- 14NS `crowd` > 14NS `familiar` > 50NS `crowd`;

VERIFIABILITY:

- 14NS `crowd` > 50NS `crowd` > 14NS `familiar`;

Annotation features experiments: MAP Results

INTERPRETABILITY:

- 14NS `crowd` > 14NS `familiar` > 50NS `crowd`;

VERIFIABILITY:

- 14NS `crowd` > 50NS `crowd` > 14NS `familiar`;
- Model size effect is most pronounced with `untargeted` & `mixed`;

Annotation features experiments: MAP Results

INTERPRETABILITY:

- 14NS `crowd` > 14NS `familiar` > 50NS `crowd`;

VERIFIABILITY:

- 14NS `crowd` > 50NS `crowd` > 14NS `familiar`;
- Model size effect is most pronounced with `untargeted` & `mixed`;
  - Unconstrained settings; larger models have more noise;

Annotation features experiments: MAP Results

INTERPRETABILITY:

- 14NS `crowd` > 14NS `familiar` > 50NS `crowd`;

VERIFIABILITY:

- 14NS `crowd` > 50NS `crowd` > 14NS `familiar`;
- Model size effect is most pronounced with `untargeted` & `mixed`;
    - Unconstrained settings; larger models have more noise;

For both INTERPRETABILITY & VERIFIABILITY:

Annotation features experiments: MAP Results

INTERPRETABILITY:

- 14NS `crowd` > 14NS `familiar` > 50NS `crowd`;

VERIFIABILITY:

- 14NS `crowd` > 50NS `crowd` > 14NS `familiar`;
- Model size effect is most pronounced with `untargeted` & `mixed`;
  - Unconstrained settings; larger models have more noise;

For both INTERPRETABILITY & VERIFIABILITY:

- `intransitives` & `ditransitives` work best with `xdx`;

Annotation features experiments: MAP Results

INTERPRETABILITY:

- 14NS `crowd` > 14NS `familiar` > 50NS `crowd`;

VERIFIABILITY:

- 14NS `crowd` > 50NS `crowd` > 14NS `familiar`;
- Model size effect is most pronounced with `untargeted` & `mixed`;
  - Unconstrained settings; larger models have more noise;

For both INTERPRETABILITY & VERIFIABILITY:

- `intransitives` & `ditransitives` work best with `xdx`;
- `transitives` work best with `ldh`;

Annotation features experiments: MAP Results

INTERPRETABILITY:

- 14NS `crowd` > 14NS `familiar` > 50NS `crowd`;

VERIFIABILITY:

- 14NS `crowd` > 50NS `crowd` > 14NS `familiar`;
- Model size effect is most pronounced with `untargeted` & `mixed`;
  - Unconstrained settings; larger models have more noise;

For both INTERPRETABILITY & VERIFIABILITY:

- `intransitives` & `ditransitives` work best with `xdx`;
- `transitives` work best with `ldh`;
  - Why? `Transitive` responses are relatively homogenous; Annotators relatively strict;

Holistic experiments

Holistic experiments

Holistic experiments use one set of 360 Spearman correlations:

targeting (2) $\times$ primacy (2) $\times$ term rep (3) $\times$ items (30) $= 360$.

Holistic experiments

Holistic experiments use one set of 360 Spearman correlations:

targeting (2) $\times$ primacy (2) $\times$ term rep (3) $\times$ items (30) $= 360$.

Holistic experiments

Holistic experiments use one set of 360 Spearman correlations:

targeting (2) $\times$ primacy (2) $\times$ term rep (3) $\times$ items (30) $=$ 360.

(`Familiar` vs. `Crowd` handled separately due to sparse data.)

Holistic experiments

Holistic experiments use one set of 360 Spearman correlations:

targeting (2) $\times$ primacy (2) $\times$ term rep (3) $\times$ items (30) $=$ 360.

(`Familiar` vs. `Crowd` handled separately due to sparse data.)

Each experiment focuses on one variable, e.g., targeting:

Divide 360 into 180 `targeted` scores and 180 `untargeted` scores; compare mean, median, etc.

Holistic experiments

Holistic experiments use one set of 360 Spearman correlations:

targeting (2) $\times$ primacy (2) $\times$ term rep (3) $\times$ items (30) $=$ 360.

(`Familiar` vs. `Crowd` handled separately due to sparse data.)

Each experiment focuses on one variable, e.g., targeting:

Divide 360 into 180 `targeted` scores and 180 `untargeted` scores; compare mean, median, etc.

SBERT uses plain text (no term rep), thus only 120 total.

Holistic experiments

Holistic experiments use one set of 360 Spearman correlations:

targeting (2) $\times$ primacy (2) $\times$ term rep (3) $\times$ items (30) $=$ 360.

(`Familiar` vs. `Crowd` handled separately due to sparse data.)

Each experiment focuses on one variable, e.g., targeting:

Divide 360 into 180 `targeted` scores and 180 `untargeted` scores; compare mean, median, etc.

SBERT uses plain text (no term rep), thus only 120 total.

(SBERT always wins over system.)

Holistic experiments: Transitivity

		intrans		trans		ditrans	
		Sys	SBERT	Sys	SBERT	Sys	SBERT
	count	120	40	120	40	120	40
14NS	mean	***0.439***	0.497	0.314	0.563	0.267	0.400
	median	**0.416**	0.479	0.304	0.555	0.276	0.444
50NS	mean	**0.423**	0.516	*0.345*	0.566	*0.278*	0.446
	median	***0.426***	0.517	*0.331*	0.561	*0.286*	0.471

Holistic experiments: `Transitivity`

		intrans		trans		ditrans	
		Sys	SBERT	Sys	SBERT	Sys	SBERT
	count	120	40	120	40	120	40
14NS	mean	***0.439***	0.497	0.314	0.563	0.267	0.400
	median	**0.416**	0.479	0.304	0.555	0.276	0.444
50NS	mean	**0.423**	0.516	*0.345*	0.566	*0.278*	0.446
	median	***0.426***	0.517	*0.331*	0.561	*0.286*	0.471

- SBERT, regardless of model size: `trans` > `intrans` > `ditrans`;

Holistic experiments: `Transitivity`

		`intrans`		`trans`		`ditrans`	
		Sys	SBERT	Sys	SBERT	Sys	SBERT
	count	120	40	120	40	120	40
14NS	mean	***0.439***	0.497	0.314	0.563	0.267	0.400
	median	**0.416**	0.479	0.304	0.555	0.276	0.444
50NS	mean	**0.423**	0.516	*0.345*	0.566	*0.278*	0.446
	median	***0.426***	0.517	*0.331*	0.561	*0.286*	0.471

- SBERT, regardless of model size: `trans` > `intrans` > `ditrans`;
- System, regardless of model size: `intrans` > `trans` > `ditrans`;

Holistic experiments: `Transitivity`

		`intrans`		`trans`		`ditrans`	
		Sys	SBERT	Sys	SBERT	Sys	SBERT
	count	120	40	120	40	120	40
14NS	mean	***0.439***	0.497	0.314	0.563	0.267	0.400
	median	**0.416**	0.479	0.304	0.555	0.276	0.444
50NS	mean	**0.423**	0.516	*0.345*	0.566	*0.278*	0.446
	median	***0.426***	0.517	*0.331*	0.561	*0.286*	0.471

- SBERT, regardless of model size: `trans` > `intrans` > `ditrans`;
- System, regardless of model size: `intrans` > `trans` > `ditrans`;
- More complex items (TTR) work best with larger models;

Holistic experiments: Transitivity

		intrans		trans		ditrans	
		Sys	SBERT	Sys	SBERT	Sys	SBERT
	count	120	40	120	40	120	40
14NS	mean	***0.439***	0.497	0.314	0.563	0.267	0.400
	median	**0.416**	0.479	0.304	0.555	0.276	0.444
50NS	mean	**0.423**	0.516	*0.345*	0.566	*0.278*	0.446
	median	***0.426***	0.517	*0.331*	0.561	*0.286*	0.471

- SBERT, regardless of model size: `trans` > `intrans` > `ditrans`;
- System, regardless of model size: `intrans` > `trans` > `ditrans`;
- More complex items (TTR) work best with larger models;
  - `trans` & `ditrans`: 50NS model is best;

Holistic experiments: `Transitivity`

		`intrans`		`trans`		`ditrans`	
		Sys	SBERT	Sys	SBERT	Sys	SBERT
	count	120	40	120	40	120	40
14NS	mean	***0.439***	0.497	0.314	0.563	0.267	0.400
	median	**0.416**	0.479	0.304	0.555	0.276	0.444
50NS	mean	**0.423**	0.516	*0.345*	0.566	*0.278*	0.446
	median	***0.426***	0.517	*0.331*	0.561	*0.286*	0.471

- SBERT, regardless of model size: `trans` > `intrans` > `ditrans`;
- System, regardless of model size: `intrans` > `trans` > `ditrans`;
- More complex items (TTR) work best with larger models;
  - `trans` & `ditrans`: 50NS model is best;
  - `intrans`: 14NS gives best median, 50NS gives best mean;

Holistic experiments: Results

Holistic experiments: Results

Targeting:

Holistic experiments: Results

Targeting:

- `targeted` > `untargeted`

Holistic experiments: Results

Targeting:

- `targeted` > `untargeted`
- 50NS models > 14NS models

Holistic experiments: Results

Targeting:

- `targeted` > `untargeted`
- 50NS models > 14NS models
  - Model size effect is most pronounced for `targeted`

Holistic experiments: Results

Targeting:

- `targeted` > `untargeted`
- 50NS models > 14NS models
  - Model size effect is most pronounced for `targeted`

Familiarity (14NS models only):

Holistic experiments: Results

Targeting:

- `targeted` > `untargeted`
- 50NS models > 14NS models
  - Model size effect is most pronounced for `targeted`

Familiarity (14NS models only):

- System: No discernible difference for `familiar` vs `crowdsourced`

Holistic experiments: Results

Targeting:

- `targeted` > `untargeted`
- 50NS models > 14NS models
  - Model size effect is most pronounced for `targeted`

Familiarity (14NS models only):

- System: No discernible difference for `familiar` vs `crowdsourced`
- SBERT: `familiar` > `crowdsourced`

Holistic experiments: Results

Targeting:

- `targeted` > `untargeted`
- 50NS models > 14NS models
  - Model size effect is most pronounced for `targeted`

Familiarity (14NS models only):

- System: No discernible difference for `familiar` vs `crowdsourced`
- SBERT: `familiar` > `crowdsourced`
  - NNS STTR < `familiar` STTR < `crowdsourced` STTR

Holistic experiments: Results

Holistic experiments: Results

Primacy:

Holistic experiments: Results

Primacy:

- System: 14NS: `primary` $<$ `mixed` (slight difference)

Holistic experiments: Results

Primacy:

- System: 14NS: `primary` $<$ `mixed` (slight difference)
- System: 50NS: `primary` $\approx$ `mixed`

Holistic experiments: Results

Primacy:

- System: 14NS: `primary` $<$ `mixed` (slight difference)
- System: 50NS: `primary` $\approx$ `mixed`
- SBERT: `primary` $>$ `mixed`

Holistic experiments: Results

Primacy:

- System: 14NS: `primary` $<$ `mixed` (slight difference)
- System: 50NS: `primary` $\approx$ `mixed`
- SBERT: `primary` $>$ `mixed`
- System & SBERT: 50NS $>$ 14NS

Holistic experiments: Results

Primacy:

- System: 14NS: `primary` < `mixed` (slight difference)
- System: 50NS: `primary` ≈ `mixed`
- SBERT: `primary` > `mixed`
- System & SBERT: 50NS > 14NS
    - System: model size effect is greatest for `primary`

Holistic experiments: Results

Primacy:

- System: 14NS: `primary` $<$ `mixed` (slight difference)
- System: 50NS: `primary` $\approx$ `mixed`
- SBERT: `primary` $>$ `mixed`
- System & SBERT: 50NS $>$ 14NS
  - System: model size effect is greatest for `primary`

Term representation:

Holistic experiments: Results

Primacy:

- System: 14NS: `primary` $<$ `mixed` (slight difference)
- System: 50NS: `primary` $\approx$ `mixed`
- SBERT: `primary` $>$ `mixed`
- System & SBERT: 50NS $>$ 14NS
  - System: model size effect is greatest for `primary`

Term representation:

- SBERT: 50NS $>$ 14NS

Holistic experiments: Results

Primacy:

- System: 14NS: `primary` < `mixed` (slight difference)
- System: 50NS: `primary` ≈ `mixed`
- SBERT: `primary` > `mixed`
- System & SBERT: 50NS > 14NS
  - System: model size effect is greatest for `primary`

Term representation:

- SBERT: 50NS > 14NS
- System: for `ldh` & `xdh`: 50NS > 14NS;

Holistic experiments: Results

Primacy:

- System: 14NS: `primary` $<$ `mixed` (slight difference)
- System: 50NS: `primary` $\approx$ `mixed`
- SBERT: `primary` $>$ `mixed`
- System & SBERT: 50NS $>$ 14NS
  - System: model size effect is greatest for `primary`

Term representation:

- SBERT: 50NS $>$ 14NS
- System: for `ldh` & `xdh`: 50NS $>$ 14NS;
  - Model size effect is greater for `ldh`

Holistic experiments: Results

Primacy:

- System: 14NS: `primary` $<$ `mixed` (slight difference)
- System: 50NS: `primary` $\approx$ `mixed`
- SBERT: `primary` $>$ `mixed`
- System & SBERT: 50NS $>$ 14NS
  - System: model size effect is greatest for `primary`

Term representation:

- SBERT: 50NS $>$ 14NS
- System: for `ldh` & `xdh`: 50NS $>$ 14NS;
  - Model size effect is greater for `ldh`
- System: for `xdx`: NS14 $>$ NS50 (very slight)

Summary

NTS: one slide

Outlook

NTS: one slide

References

Sampling data: Response length

	n=14		n=50	n=70
	Fam	Crowd	Crowd	NNS
Intrans	5.5	4.9	4.9	4.9
Trans	6.9	6.3	6.2	6.7
Ditrans	7.8	7.2	7.2	8.3
Target	6.5	5.4	5.4	6.3
Untarg	6.9	6.8	6.8	6.9
primary	N/A	5.7	5.8	6.6
mixed	6.7	6.5	6.4	N/A
Total	6.7	6.1	6.1	6.6

Table: Comparing average response length (in words) for the samples used throughout this chapter as NS models and NNS test sets, in total and by parameter setting.