Semantic Analysis of Image-Based Learner Sentences

Levi King
Indiana University

August 6, 2021

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 rely on outdated, ineffective methods: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

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

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

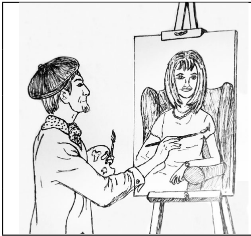
	**Response (L1)**
	He is droning his wife pitcher. (Ar)
	The artist is drawing a pretty women. (Ch)
	The artist is painting a portrait of a lady. (En)
	The painter is painting a woman's paint. (Sp)

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

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

Pilot Study Processing

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%
  - I.e., *Acceptable* covered, *unacceptable* not covered;

Pilot Study Processing

Semantic similarity scoring

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

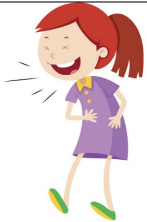
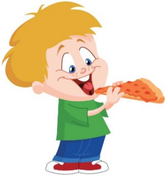
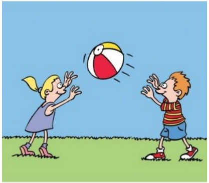
NNS response score = cosine distance of NS and NNS tf-idf scores;

- 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;
- Mean Average Precision (MAP) *acceptable* responses: ≈51%
- Process is more robust & generalizable;
- Dataset (especially NS models) and annotation are weak;

Data collection

I chose to use picture description task (PDT) data because:

- Most ICALL applications rely on images;
- Simple images constrain the range of likely responses.

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

- 499 participants: 358 NS (crowdsourced), 141 NNS (ESL students)
- 30 items: Roughly 300 NS responses & 140 NNS responses per item

Data collection

NTS:
Explain:

- targeted vs untargeted
- familiar vs crowdsourced
- first vs second responses

Describe:

- participants, resulting dataset: total responses, demographics

Annotation Features

Feature requirements:

- *reliablity*: consistently annotated by multiple humans;
- *validity*: directly testing for the desired constructs;

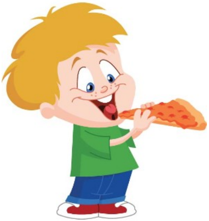
First iteration: **accuracy (A)** & **native-likeness (NL)**

- **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*?

Several iterations later, 5 binary features:

- **Core event**: captures main action
- **Answerhood**: directly answers prompt
- **Grammaticality**: no grammar problems
- **Interpretability**: evokes clear mental image
- **Verifiability**: info is 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).

Feature reliability

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

- Features do not contribute equally to response “goodness”
- Raters perform holistic preference test (with annotations hidden)

What is the boy doing?	Pref?	C	A	G	I	V
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

Preference reliability (feature weights)

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

Table: Preference task agreement scores for two annotators on a sample of 300 response pairs; expected chance agreement, observed agreement and Cohen's Kappa.

Weighted annotation scores

- Calculate weighted annotation score (S_{wa}) for each NNS response;
- Rank by S_{wa} ($\rightarrow R_{wa}$); use this as gold standard or benchmark;
  - Score system output: R_{wa} vs. system ranking $\rightarrow$ Spearman ρ

What is the boy doing?	C	A	G	I	V	S_{wa}	R_{wa}
The boy is eating.	0	1	1	1	1	0.635	4
A baby is eating pizza	0	0	1	1	0	0.279	5
The boy enjoys his pizza.	1	0	1	1	1	0.907	2
the boy is eating pizza	1	1	1	1	1	1.0	1
The kid is eats pizza	1	0	0	1	1	0.852	3

Analyzing NNS responses

At this point, my goal is a system that scores and ranks NNS responses via comparison with the crowdsourced NS responses. The system produced ranking should correlate highly with the R_{wa}.

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

Analyzing NNS responses

The system works like this; for each item:

Generate a NS model:

1. dependency parse the collection of NS responses;
2. get tf-idf score for each unique dependency (via a large balanced corpus).

Score each NNS response:

1. As above: dependency parse, tf-idf;
2. Compare NS vs NNS tf-idf vectors: 1 - cosine = response score.

Finally, the NNS responses are ranked by score, and the Spearman rank correlation between R_{wa} and the system is taken as the system configuration score for the item.

By selecting different parameter settings in this approach, I arrive at 12 different system configurations. Each configuration scores and ranks all NNS responses.

System configurations

Consider this simplified set of 2 parameters *x* 2 settings = 4 configurations.

- **Dependency format**:
  - **labeled**: e.g., nsubj(eat,boy); nobj(eat,pizza)
  - **unlabeled**: e.g., ⟨null⟩(eat,boy); ⟨null⟩(eat,pizza)
- **NS response model**: Note: Each NS participant gave *two* responses per PDT item
  - **first**: Model contains only the first response from NS
  - **mixed**: Model is half first reponses and half second responses

dep\model	first	mixed
labeled	lab_first	lab_mixed
unlabeled	unlab_first	unlab_mixed

System configurations

- Score & rank NNS responses using different configurations;
- Compare with R_{wa} to get a Spearman correlation.

NNS	S_{wa}	R_{wa}	S_{lf}	R_{lf}	S_{uf}	R_{uf}
p1	0.63	4	0.53	4	0.11	5
p2	0.27	5	0.13	5	0.15	4
p3	0.90	2	0.91	1	0.68	1
p4	1.0	1	0.80	2	0.41	2
p5	0.85	3	0.77	3	0.20	3
Spearman ρ			.899		.799	
Spearman p-val			.037		.104	

- *lf* is *labeled_first*; *uf* is *unlabeled_first*:
- *labeled* for labeled dependencies (vs. *unlabeled*)
- *first* for models containing only the first response from NS (vs a *mix* of first and second responses)

Results & Optimization

NTS: Multiple slides; describe most salient findings

Summary

NTS: one slide

Outlook

NTS: one slide

References