Leveraging Known Semantics for Spelling Correction

Levi King and *Markus Dickinson*

Indiana University
Bloomington, IN USA

leviking@indiana.edu, md7@indiana.edu

ABSTRACT

Focusing on applications for analyzing learner language which evaluate semantic appropriateness and accuracy, we build from previous work which modeled some aspects of interaction, namely a picture description task (PDT), with the goal of integrating a spelling correction component in this context. After parsing a sentence and extracting semantic relations, a surprising number of analysis failures stem from misspellings, deviating from expected input in ways that can be modeled when the content of the interaction is known. We thus explore the use of spelling correction tools and language modeling to correct misspellings that often lead to errors in obtaining semantic forms, and we show that such tools can significantly reduce the number of unanalyzable cases. The work is useful for any context where image descriptions or some expected content is available, but not necessarily expected linguistic forms.

KEYWORDS: picture description task, semantic analysis, spelling correction, language modeling.

Levi King and Markus Dickinson 2014. Leveraging known semantics for spelling correction. *Proceedings of the third workshop on NLP for computer-assisted language learning.* NEALT Proceedings Series 22 / Linköping Electronic Conference Proceedings 107: 43–58.

1 Motivation

Much current work on analyzing learner language focuses on grammatical error detection and correction (e.g., Dale et al., 2012) and less on semantic analysis; many Intelligent Computer-Assisted Language Learning (ICALL) and Intelligent Language Tutoring (ILT) systems (e.g., Heift and Schulze, 2007; Meurers, 2012) also focus more on grammatical feedback. An exception to this rule is *Herr Komissar*, an ILT for German learners that includes rather robust content analysis and sentence generation (DeSmedt, 1995), but this involves a great deal of hand-built tools and does not connect to modern NLP. Some work addresses content assessment for short answer tasks (Meurers et al., 2011), but there is still a need to move towards naturalistic, more conversational interactions (see Petersen, 2010). Such interactions are both more and less difficult to process: to provide feedback requires keeping track of the content of the interaction, but such content can also be used to disambiguate new learner productions. We exploit this tension in the context of spelling correction, as semantic information severely restricts the learner's expected content, and thus also their word forms.

Since our overarching goal is to move towards the facilitation of ILTs and language assessment tools that maximize free interaction, we have to deal with removing impediments to interaction. Given the preponderance of spelling errors in learner data, and specifically interactive data (King and Dickinson, 2013), our specific goal is to use basic NLP (pre)processing—namely, language modeling for spelling correction—to make the meaning of a learner's sentence clearer. We examine methods for automatically correcting misspellings, showing that preprocessing with spelling correction tools, when information about the interactive context is known (i.e., the picture's description), can greatly reduce downstream errors.

This may seem like a niche problem, but: 1) spelling errors are generally a major problem in analyzing learner data (Leacock et al., 2010; Flor et al., 2013); 2) the specific focus we have right now, on picture description tasks (PDTs), connects not only with a desire for more interactive tools, but also for language assessment (Somasundaran and Chodorow, 2014); and 3) our work seeks to unpack the connection between relatively "shallow" errors, namely spelling errors, with "deeper" errors, namely semantic ones. Unlike, for example, linguistic abstractions such as part-of-speech, both are intimately rooted in the particular lexical items used. This then raises the question of whether we are modeling what the learner said (modulo some spelling variation), what the learner intended, or what the learner should have intended, an issue we take up in section 4, after covering the background in section 3. The methods are covered in section 5 and the evaluation in section 6.

2 Related Work

Research into the patterns of spelling errors particular to native speakers (NSs) and non-native speakers (NNSs) highlights the challenge of applying spelling correction techniques to non-native text. Flor et al. (2013) examined spelling errors found in the ETS Spelling Corpus (3000 GRE and TOEFL essays) and found that NNS spelling errors were more severe (i.e., had a greater edit distance from the intended word) than NS errors. Moreover, NNSs made more spelling errors than NSs for words of 3-7 letters, but this trend reversed for words of 8 letters or more. These effects were shown to disappear among the most proficient NNSs in the sample, however. Similarly, Hovermale (2010) compared the spelling errors in corpora of Japanese learners of English to previous studies of NS spelling errors and found that the learner errors have a greater average edit distance and are nearly twice as likely to involve the first letter of the word. Given such variability in form, correcting spelling errors for NNSs strictly via edit

distance operations would thus seem to have its limits.

Using the ETS Spelling Corpus and the ConSpell spelling correction tool, Flor (2012) demonstrates significant gains in automatic spelling correction when modules using contextual information are added. Four types of context, each of which benefitted spelling correction, were explored: 1) word *n*-grams (length 1–5) and a web-scale language model (LM); 2) word *n*-grams and the positive normalized pointwise mutual information (PNPMI) of the words within them (based on a web-scale distributional model); 3) the entire essay (and the recurrence or lack of a given candidate spelling correction in the essay); and 4) the text of the essay prompt. Notably, a 3.8% improvement comes through the use of "global mutual optimization", i.e., at each given spelling correction decision, the module is biased not only toward other words in the text, but also the candidate spelling lists of these other words. The work presents a strong case for the use of *n*-grams with both LMs and PNPMI, as the best results come from this setting, boosting performance 11.48% above the non-contextual spelling correction baseline.

Flor and Futagi (2012) further examine the use of context for correcting learner misspellings and claim that three major issues contribute to the task's difficulty: "local error density" (a misspelled word near other misspellings) weakens *n*-gram approaches; poor grammar can lead to the selection of an incorrect spelling candidate based on its agreement with nearby incorrect words; and competition among closely related spelling candidates can lead to the selection of an incorrect inflectional variant. These challenges indicate that for potentially error-rich learner sentences, sentence or *n*-gram level contexts may be more effective when combined with higher-level contextual information, such as task prompts and discourse-level information about verb inflections. We explore including information about picture content.

3 Background

3.1 Data

In previous work (King and Dickinson, 2013), we collected responses to a picture description (PDT) task to approximate interactive behavior. The current study relies on the same set of responses. We use a PDT because it helps constrain both form and content, without providing textual prompts that may influence a learner. Moreover, PDTs are a well-established tool in areas of study ranging from SLA to Alzheimer's disease (Ellis, 2000; Forbes-McKay and Venneri, 2005). The use of visual stimuli also helps model the visual nature of online games. The stimuli are chosen to elicit relatively unambiguous transitive sentences.

The PDT consisted of 10 items (8 line drawings and 2 photographs) intended to elicit a single sentence each; an example is given in Figure 1. Participants were asked to view the image and describe the action in a complete

Response (L1)
The man killing the beard. (Arabic)
A man is shutting a bird. (Chinese)
A man is shooting a bird. (English)
The man shouted the bird. (Spanish)

Figure 1: Example item and responses

sentence, with any tense or aspect appropriate. 25 of the 39 non-native speaker (NNS) participants performed the task in a setting where automatic spell checking was disabled; the remaining 14 performed the task online on their own computers, and although they were instructed to disable spell checking, we have no way of knowing if they did so.

The NNSs were intermediate and upper-level adult English learners in an intensive English as a Second Language program at Indiana University. This data set contains responses from 53 informants, including native speakers (NSs) (14 NSs, 39 NNSs), for a total of 530 sentences. The distribution of first languages (L1s) is: 16 Arabic, 7 Chinese, 14 English, 2 Japanese, 4 Korean, 1 Kurdish, 1 Polish, 2 Portuguese, and 6 Spanish.

3.2 Method

As in King and Dickinson (2013), our method to obtain a semantic form from a NNS production takes two steps: 1) obtain a syntactic dependency representation from the off-the-shelf Stanford parser (de Marneffe et al., 2006; Klein and Manning, 2003), and 2) obtain a semantic form from the parse, via a small set of hand-written rules. To illustrate this process, consider (1). This sentence is passed through the parser to obtain the dependency parse shown in Figure 2. Based on the presence of the `nsubjpass` (noun subject, passive) node, the extraction script takes the logical subject from under the `agent` label, the verb from `root`, and the logical object from `nsubjpass`. This results in the semantic triple *shot(man,bird)*, lemmatized to *shoot(man,bird)*, using the Stanford CoreNLP lemmatizer (Manning et al., 2014). Very little effort is needed: the parser is pre-built; the decision tree is small; and the extraction rules are minimal. Note, too, that certain relations (e.g., `det`) are completely ignored in the extraction.

(1) A bird is shot by a man.

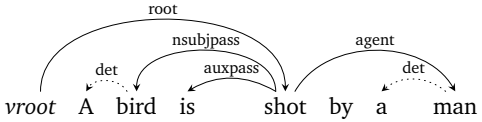

Figure 2: The dependency parse of (1)

One is able to use little effort in part due to the constraints in the pictures. For figure 1, for example, *the artist*, *the man in the beret*, and *the man* are all acceptable subjects, whereas if there were multiple men in the picture, *the man* would not be specific enough.

Evaluation in King and Dickinson (2013) addresses two major questions. First, how accurate is the extraction of semantic information from potentially innovative sentences? Secondly, how much coverage does one have in a gold standard of semantic forms (triples), to capture the variability in meaning in learner sentences? We focus more on the first question and again use native speaker semantic forms as a proxy for a gold standard—albeit, limited by mismatches between native and (correct) non-native ways of saying the same thing. To mitigate this and better see the effect of spelling correction, much of our evaluation relies on hand-analysis which determines whether a "reasonable gold standard" could contain the information (see section 4).

Semantic extraction For the purpose of evaluating an extraction system, King and Dickinson (2013) define two major classes of errors. The first are *triple errors*, responses for which the

system fails to extract one or more of the desired subject, verb, or object, based on the sentence at hand and without regard to the target content. Second are *content errors*, responses for which the system extracts the desired subject, verb and object, but the resulting triple does not accurately describe the image (i.e., is an error of the participant's). In this paper, we focus on reducing the triple errors, i.e., system errors. For example, the spelling error in (2) leads to a completely incorrect triple. We will unpack our error types in section 4.

(2) A man swipped leaves. ⇒ leave(swipped,man)

Focusing on triple (system) errors, we have obtained 92.3% accuracy on extraction for NNS data and roughly the same for NS data, 92.9% (King and Dickinson, 2013). Furthermore, more than half of the errors for NNSs involve misspellings (4.1% of the total 7.7% of errors). For a system interacting with learners, spelling errors are thus a high priority (cf. Hovermale, 2008).

Content errors are subcategorized as *spelling* or *meaning* errors, depending on whether the resulting triple has spelling errors that do not result in real words—as in (3)—or that do result in real but unintended words and thus convey an inappropriate meaning (e.g., *shout(man,bird)* instead of *shoot(man,bird)*). We will see this distinction play out in the spelling correction techniques in section 5.

(3) The artiest is drawing a portret. ⇒ drawing(artiest,portret)

Approximately 15% of NNS triples are content errors (King and Dickinson, 2013). These cases are ones for which the learner needs feedback, but there are two barriers in providing feedback: 1) without fixing the triple errors, they will be automatically grouped into the content error cases, since they do not match the gold standard; and 2) even if one knows something is an error, to obtain feedback one would ideally know the target the learner was (or should have been) aiming for. Our approach to spelling correction addresses both of these concerns by cleaning up the misspelled cases—including many of the "content" errors rooted in misspellings.

Semantic coverage In King and Dickinson (2013), we take a set of native speaker (NS) responses for the same PDTs as the gold standard, garnering coverage numbers around 25% for types and 50% for tokens—i.e., about half of *correct* NNS responses are not in the gold standard. Since our focus is on improving accuracy, we use the same gold standard, but augment the analysis with hand-evaluation of whether a response should have been in the gold standard (section 4). Still, with spelling modifications being made to make a NNS response more native-like, we may be able to increase coverage, i.e., to find a (gold) NS triple that matches.

4 Error Types

As alluded to above, our goal is to model a close intended meaning of every NNS sentence, in order to provide a platform for providing feedback. By *close intended meaning*, we mean, *a meaning that matches some correct answer and whose corresponding form is a reasonable distance to what the NNS wrote*. Since we have not interviewed participants with follow-up questions about the intention of their responses and cannot assume a follow-up in the general case, we take the close intended meaning as the meaning they should have intended, given their production.

For evaluation, then, we want to measure the extent to which we are able to take a NNS form and produce a plausible target meaning for their "intention," i.e., a viable semantic triple. The evaluation should answer: 1) Is there a valid meaning? and, if not, 2) what step in the process prevented a valid meaning from being derived?

The entire system is outlined in section 5.2 (see Figure 3), but essentially we have this pipeline:

0. Sentence produced by NNS
1. (optional) Spelling corrector to generate close intended form
2. Syntactic parser to obtain word-word relations
3. Semantic extractor to obtain semantic forms
4. Comparison to gold standard

Starting with the gold standard comparison and working backwards, we evaluate at every step:

1. Is the triple covered by the gold standard? **Yes:** Not an error. **No:** Continue to next step.
2. Should the triple be covered by a reasonable gold standard? **Yes:** *Gold* miss ("error"). **No:** Continue to next step.
    - Given the limited coverage of the NS gold standard, we add this manual step, so as not to focus too much on one particular gold standard.
3. Is the form (either the NNS sentence or the close intended meaning chosen by the correction module) well-formed and appropriate for the item but the extracted triple is not covered by a reasonable gold standard? **Yes:** *Triple* error. (These could be subcategorized as parser, lemmatizer, or extractor errors.) **No:** *Form* error.

Note that for our purposes here, a "good" triple should indicate an appropriate subject, verb and object, whether directly or indirectly. In most cases, this is a complete *V(S,O)* triple. For some concepts, however, a verb may imply its object, or vice versa. Item 3 of the PDT, for example, shows a woman riding a bicycle. This could be represented as a transitive action, resulting in a triple like *ride(woman,bicycle)*. However, this could also be construed as an intransitive, such as *cycle(woman,NONE)*. Both of these triples should be considered appropriate. A form like *A woman is on a bicycle* should also be considered appropriate, because the obvious action involving a person on a bicycle is *ride* (or *pedal*, *travel*, etc.), even though the extracted triple (*be(woman,bicycle)*) is less descriptive. An intransitive resulting in a triple like *ride(woman,NONE)* is inadequate, however, because *ride* does not sufficiently imply the object.

Similar cases occur among the responses to item 9, which shows two boys rowing a boat. We might consider *row(boy,boat)* to be an ideal triple for this item, but we also accept *be(boy,boat)* here, as in *Two boys are on a boat*. Note that this is only acceptable because in the absence of more detail, a reasonable person given the information that some human is performing some action on or involving a boat would likely assume that the action involves using the boat for its intended purpose—to travel on water, and that could be represented with a more specific verb. We should also accept *row(boy,NONE)* here, because (unlike *ride*) the verb *row* sufficiently implies its object (a boat). Similarly, *boat(boy,NONE)* is adequate, because as a verb, *boat* indicates both the presence of a boat and the action of riding the boat.

5 Spelling Correction Modifications

5.1 Motivation for spelling correction via language modeling

The initial approach to this task in King and Dickinson (2013) revealed that the ability of the system to recognize NNS responses as correct was often hindered by minor errors in spelling. Misspellings are especially problematic here because they can derail the semantic evaluation of a response by leading to errors in the syntactic interpretation of the sentence. Whereas human listeners or readers can use the context and their knowledge of the language to infer the intended pronunciation or spelling of a mispronounced or misspelled word, the initial approach lacked any such compensatory strategies. To improve the system's ability to handle NNS data, we implement a spelling correction module, which we see as an attempt to endow the system with some of the general language knowledge that a human would use upon encountering a misspelling. Importantly, we incorporate contextual information about the picture by giving this module access to NS responses (i.e., picture descriptions), allowing it to prefer corrected spellings that may be relevant to the context.

We begin with a context-independent spelling corrector, Aspell (Atkinson, 1998), but on finding mixed results with only this basic spelling correction module—due to its lack of incorporation of context—we expand the process to include a statistical language model (LM) based on word trigrams (section 5.2). The *n*-gram LM essentially takes a large body of English text, counts the occurrences of each sequence of *n* words, converts these counts to relative frequencies, and uses these relative frequencies to calculate the probability of new texts. The LM has the effect of evaluating the likelihood of multiple possible spellings for a misspelled word (as provided by a context-independent spelling correction module) in the context of surrounding words. In this way, we further attempt to use contextual information and general knowledge of the language to model the close intended meaning while overlooking minor errors in orthography.

The implementation of these tools raises some questions about about how fair or appropriate it is to try to estimate a learner's intended utterance, and just exactly what spelling correction is and is not (or should and should not be). Any automatic or manual approach at correcting malformed learner language or interpreting its meaning encounters ambiguous and challenging cases. This is why we defined our goal in section 4 as that of deriving a close intended form, essentially sidestepping the question of what the ultimate correction *means* and instead focusing on what the correction can tell us about the linguistic utterance's relation to the picture.

This goal, it should be pointed out, is in keeping with a prioritization on encouraging learners to produce more language and on interaction with learners, as opposed to prioritizing grammatical or orthographic perfection. Deriving a close intended form should be able to inform a system towards an appropriate piece of feedback. This can also be seen as "giving the benefit of the doubt" to learners, finding the gold item that looks close; giving the benefit of the doubt is particularly true in the joint evaluation described in Section 6.2, where we consider each original response as well as its corrected version.

5.2 Spelling correction process

Aspell In our first attempt at spelling correction, we added a preprocessing step using Aspell, a spelling correction tool (Atkinson, 1998). For each PDT item, the NNS sentences were passed through Aspell. Words recognized by Aspell were not changed. For words that Aspell considered misspelled, the ranked list of Aspell suggestions was compared with a list of words used in the

NS sentences. The highest ranked suggestion that was also in the NS word list was accepted as the corrected spelling. If no match was found, the first suggested word was accepted, and the sentences were then passed to the rest of the pipeline. A major limitation of this correction was the fact that misspellings resulting in real words were not addressed. For example, several participants responded to one item with the real word *shout* but clearly intended *shoot* (cf. *A man shoots a bird*). Indeed, evaluation of this simple Aspell approach revealed that it introduced significantly more errors than it corrected. Thus, we omit this method from further discussion and focus on a more contextually informed approach incorporating language modeling.

LM pipeline In the approach discussed hereafter (the *LM pipeline*), Aspell (via the Enchant python package (Lachowicz, 2003)) is used to obtain a list of spelling suggestions for all words, including those that appear to be properly spelled. These candidate spellings are combined to form a list of candidate sentences for each response. Each candidate sentence is then compared with an *n*-gram language model to obtain a perplexity score—i.e., a measure of how likely the sentence is, given the LM. The candidate sentence with the lowest perplexity is chosen automatically as the best correction. A diagram of the entire semantic extraction process incorporating the spelling correction and language modeling tools is given in Figure 3.

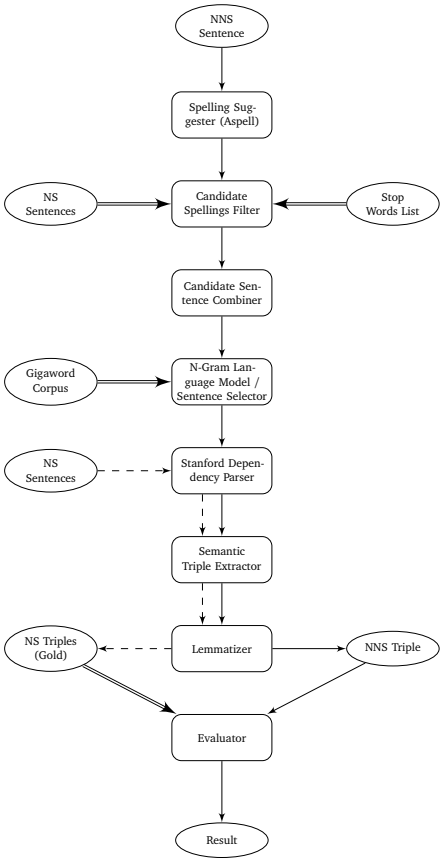

Figure 3: Semantic triple evaluation pipeline. Boxes are system components, circles are data; double arrows indicate training, dashed arrows show the obtaining of gold standard triples.

The computational costs of this approach have the potential to be very great. The number of spelling suggestions for a given word range between zero—for egregious misspellings of long words, unlike anything in the dictionary—and up to 50, for words within a short edit distance of known words (e.g., *pet*). The average number of suggestions for the words in the NNS responses is roughly 31. The average sentence length among the entire data set of NNS responses is 7.2 words. This would result in approximately $31^{7.2}$ (nearly 55 billion) candidate sentences for a single NNS response. We took several steps to prune the number of candidate words and sentences in order to make this process more manageable.

For this pruning, we draw on the NS responses; this decision is based on the assumptions that NS responses are correct and that the PDT constrains the content of responses. We

create a NS word list by taking every word form found in the NS responses. We also use a list of stop words consisting of the 200 most common English words, to filter out short function words that would create too many candidates.

This process of forming candidate sentences for a NNS response assumes that while there may be misspellings, the number of words in the sentence is fixed. That is, a word may be replaced by another word, but no word may be removed and no additional words may be inserted. (Rare exceptions may occur when the spelling correction tool suggests that an unrecognized word be split into two words.) This is a limitation of the current implementation and should be addressed in the future, perhaps incorporating techniques for word normalization over word lattices from the speech recognition literature, such as those in Sproat et al. (2001).

For a given NNS response in this pipeline, each token is given a status of `fixed` or `unfixed`. Each word enters the pipeline as `unfixed`; it is then compared with the stop words list, and if a match is found, the status is changed to `fixed`. The remaining `unfixed` words are then compared with the NS word list and again, matches are `fixed`. For any token with a `fixed` status, no candidate spelling corrections will be considered. Thus we assume that a NNS word that matches a stop word is correct, as English learners at this level are unlikely to misspell common function words. We also assume, given the constraints of the PDT, that a NNS word that matches a NS word is correct.

Next, we handle misspellings where no sentential context is needed, given the contents of the picture. Each `unfixed` word is passed to Enchant and a list of candidate spellings is obtained. Note that a ranked list of spelling suggestions is generated even for words that appear to be properly spelled. This list is compared with the NS list; if one or more matches are found, the highest ranked candidate word is selected, and the status is `fixed`. If no match is found, the status remains `unfixed`, and the entire list of candidate words is added to that word position.

After that pruning, a list of candidate sentences can now be generated by iterating through candidate words for `unfixed` positions to generate every possible combination with all the `fixed` words. As mentioned above, NNS responses in the data set contain an average of 7.2 words, and at this stage, 6.5 words are `fixed` and 0.7 words are `unfixed`, resulting in an average of $31^{0.7}$ (roughly 11) candidate sentences per NNS response, drastically reducing the computational costs of the remaining steps in the pipeline. Many well-formed responses result in no candidate sentences beyond the original form, while the largest number of candidates seen among the entire set was 57,300, for a 10-word sentence.

For each NNS sentence, the original sentence and its list of candidates are passed to the language model for evaluation. Here we use the CMU Statistical Language Modeling Toolkit (Clarkson and Rosenfeld, 1997) in a trigram setting trained on a sample of the English Gigaword Corpus (Graff et al., 2007) containing roughly 250 million words in 10 million sentences of newspaper text. The candidate sentences are ranked according to their perplexity with regard to the language model. The sentence with lowest perplexity is selected as the most likely sentence and passed through the remaining steps of the pipeline, as shown in the lower half of Figure 3.

The source of PDT descriptions We use the NS responses here as our proxy for a description of the picture content. This is distinct from using the NS responses as a gold standard to compare the final triples against, as in King and Dickinson (2013); indeed, this is why we manually check triples in this work, assuring that we truly know whether a triple is valid or not. In either case, we will see the limitations of using NS responses for these purposes.

6 Spelling Correction Evaluation

Here we present the results of the modifications detailed above. At this stage, we are primarily interested in our system's ability to robustly extract evaluable triples, potentially in the face of minor errors. While we present coverage scores in the following sections—calculating coverage with respect to the particular (and limited) gold standard set of triples—we focus mainly on the effect the modifications have on (Form) error counts.

	NNS	LM	Joint(NNS)	Joint(LM)	Joint(Oracle)
Coverage	134	149	152	152	152
Gold Misses	125	110	125	109	141
Triple Errors	13	13	13	14	15
Form Errors	118	118	100	115	82
Total Form/Triple Errors	131	131	113	129	97

Table 1: Errors types and coverage for the full set of responses (390 sentences). *Joint* indicates a joint analysis of both sources (NNS & LM); the default source in parentheses was chosen in cases where neither triple was found (see Section 6.2).

6.1 LM pipeline errors

With no attempt at spelling correction, the 390 NNS responses result in a total of 131 true errors, with an additional 125 misses due to an incomplete gold standard and coverage of 134 non-errors, as seen in Table 1. For our evaluation, we are most concerned with reducing the Form errors—which may result in more Gold misses, depending upon whether a valid triple is in the gold standard or not.

Evaluating the LM output results in reducing the number of Gold misses by 8.3%, from 125 to 110, with the Triple and Form error counts unchanged. But this does not tell the full story of changes. If we look closer, as in the first column of Table 2, we see that in comparing the LM triples to the NNS triples, a total of 73 responses change from one error type to another. This includes the conversion of 18 Form errors to non-errors (∅) and three non-errors to Form errors. An example of a "recovered" Form error can be seen in (4). In this case, *shoots* and *bird* are both present in the NS gold standard responses, which helps the LM obtain an acceptable triple.

Change		LM	J(NNS)	J(O)
∅ ↦ ∅	↔	131	149	152
∅ ↦ Gold	↓	0	0	0
∅ ↦ Trip.	↓	0	0	0
∅ ↦ Form	↓	3	0	0
Gold ↦ ∅	↑	0	0	0
Gold ↦ Gold	↔	93	94	125
Gold ↦ Trip.	↓	1	0	0
Gold ↦ Form	↓	31	15	0
Trip. ↦ ∅	↑	0	0	0
Trip. ↦ Gold	↑	0	1	0
Trip. ↦ Trip.	↔	11	11	13
Trip. ↦ Form	↓	2	2	0
Form ↦ ∅	↑	18	3	0
Form ↦ Gold	↑	16	30	16
Form ↦ Trip.	↑	2	2	2
Form ↦ Form	↔	82	83	82
Changed		73	53	18
Unchanged		317	337	372
Total	↔	317	337	372
Total	↑	36	36	18
Total	↓	37	17	0

Table 2: The number of changes between error types, moving from NNS to LM, from LM to J(NNS), and from J(NNS) to J(Oracle).

(4) a. NNS: the old man shouts to beird. ⇒ NONE(shout,NONE)
b. LM: the old man shoots to bird. ⇒ shoot(man,bird)

An example of a Form error introduced by the LM pipeline is seen in (5). Here we see that given the NNS sentence, the lemmatizer was robust enough to properly arrive at *shoot* from the misspelled *shooted*. While both *shoot* and *shot* were among the 12 words suggested by Aspell to replace *shooted*, the LM preferred *shouted*.

(5) a. NNS: a man shooted a bird. ⇒ shoot(man,bird)
b. LM: a man shouted a bird. ⇒ shout(man,bird)

Additionally, the LM pipeline changes 31 Gold "errors" to Form errors. While this does not affect coverage or the total error counts when evaluated under the current gold standard, these cases are clearly problematic. One example is shown in (6b), modified by the LM pipeline from (6a).

(6) a. NNS: a person was cutting fruit. ⇒ cut(person,fruit)
b. LM: a person was cutting fraud. ⇒ cut(person,fraud)

The issue stems from the fact that the LM was trained on newspaper text, leading it to prefer words and phrases prevalent in the news, (*cutting fraud*), while giving higher perplexity to those less common in news stories (*cutting fruit*). Other examples of this domain-based over-correction include the changing of *biking* to *backing*, *cleaning* to *learning*, and *chopping* to *shipping*. Besides choosing better LM training data, future work could use other methods of analysis to avoid these problems. For example, using WordNet (Fellbaum, 1998) to discover that "fruit" is a hypernym of "apple" (the object described by all NSs), and thus (possibly) acceptable, would eliminate the need to process some spelling candidates via the LM.

Due to the design of the LM pipeline, these problems are compounded by the sparseness of the gold standard. The NS responses are used to derive the gold standard, but also to derive a list of context-appropriate words for each item. As described above, this word list is used to select appropriate spelling candidates from the correction tool before the recombined sentences are evaluated by the LM. The over-correction problem is exacerbated when the PDT item depicts an action for which NSs know a specific word but NNSs may not, like *raking* or *rowing*. These items highlight the disadvantages of relying on NS responses. For such prompts, we observe: a) relatively high numbers of candidate sentences, because fewer candidate spellings are decided by the NS word list, as well as: b) higher numbers of Form errors, because we shift the burden of deciding contextually appropriate words to the LM.

6.2 Joint Evaluation

So far, each sentence form and triple output by the LM pipeline is evaluated alone, without regard to the NNS form or the output of the the original process. In this section we present a joint analysis, wherein we take both the LM triple and its NNS counterpart; in cases where one of the two triples is found in the set of NS responses, we keep that triple and ignore the other; in cases where neither triple is found, we default to the NNS triple; we refer to this as *Joint(NNS)*. (A joint analysis defaulting to the LM (*Joint(LM)*) was also performed, but this resulted in weaker performance, as shown in Table 1, and is omitted from the discussion). The

idea behind this joint analysis is simply to give the system the choice between two triples for a single response, using information about the picture's contents (NS responses) to pick one, effectively allowing us to undo any errors introduced by Aspell or the LM.

We again focus primarily on the changes in error counts. Unlike the analyses above, however, under this joint evaluation there is an unavoidable possibility for the set of NS responses to affect error counts. This is because a triple's presence or absence in this set determines which of the two triple versions is considered. Consider the following constructed example to illustrate this concern. Under our joint analysis, given an original triple of *shout(hunter,bird)*, which is an error (and of course absent from the list) and an LM triple of *shoot(hunter,bird)*, which is correct but absent from the NS list, we default to the original triple, thus including an error that would have been avoided if the NS list had covered the LM triple.

Such cases illustrate the fact that the error types are not equally (un)desirable. A Gold miss is better than a Form or Triple error for us, because the Gold miss is not a system error at all and could be covered by an improved gold standard. Likewise, a (NNS) Form error changed to a (LM) Triple error is a partial success, because this means the spelling correction module was successful, while the parser or semantic extractor needs improvement. To address these issues, we perform a *Joint(Oracle)* experiment (section 6.2.2), in which errors were ranked by preference, from non-error, to Gold miss, to Triple error, to Form error. In cases where neither the NNS nor the LM triple was found and the error types were different, the oracle chooses the preferred error type, minimizing Form errors and maximizing Gold misses. The results of this experiment give a better approximation of the potential of the current system given an ideal set of triples covering the content of the picture (which the NS responses serve as a proxy for).

6.2.1 Joint(NNS) errors

The Joint(NNS) experiment gives coverage to 152 triples, and compared with the LM pipeline, it results in a net reduction of 18 Form/Triple errors, from 131 to 113. While the error type was changed for 53 responses, this improvement is partly the result of three Form errors converting to non-errors (Table 2). An example of such a gain is seen in the NNS response (5a) and the LM version (5b); this time, as the LM triple is not found, we default to the correct NNS triple, undoing the error introduced by the LM pipeline. Another example of an LM error avoided under the joint analysis is shown in (7b), as it defaults to the NNS response seen in (7a).

(7) a. NNS: a boy is playing a soccer alone. ⇒ play(boy,soccer)
b. LM: a boy is playing a soccer one. ⇒ play(boy,one)

Importantly, we also see 30 Form errors from the LM model become Gold misses (Table 2), leading to an overall reduction of Form errors from 118 to 100 (Table 1). This Joint(NNS) model, then, is doing exactly what it is designed to do: removing Form errors by changing the spelling into something with a valid semantics.

6.2.2 Joint(Oracle) errors

As mentioned in section 6.2, many positive changes introduced by the LM pipeline are not fully realized under the LM or Joint(NNS) experiments. We investigate that here by using an oracle to choose the preferred error type in cases where neither triple is found. As a result, 18 correct changes introduced by the LM—but ignored by defaulting to the NNS under the Joint(NNS)

setting—are retained under the Joint(Oracle) setting. These include two Form errors converted to Triple errors and 16 Form errors converted to Gold misses (Table 2). Note that coverage remains at 152 (Table 1).

An example of a Form error converted to a Triple error can be seen in (8b), the form and triple derived from the NNS response in (8a). We consider (8a) to be a Form error, because *cycle* does not fully describe a bicycle. In (8b), despite the fact that the LM pipeline made an appropriate correction and returned a perfectly acceptable form, the derived triple is incorrect. This is the result of an inappropriate parse, with *rides* given a plural noun (NNS) part of speech tag and *the woman rides* labeled as a noun phrase.

(8) a. NNS: the woman rides on her cycle. ⇒ NONE(ride,cycle)
b. LM: the woman rides on her bicycle. ⇒ NONE(ride,bicycle)

This kind of error is representative of a pattern among the Triple errors found across the dataset: third person present tense verbs are regularly analyzed (via the parser's built-in part-of-speech tagger) as plural nouns, leading to the extraction of an incorrect triple. This is seen for *rides, boats, rows,* and *paints*. A game setting would likely alleviate this problem by constraining responses to the past tense, but NNSs may also need to be reminded that the simple present is usually reserved for describing general truths.

An example from the 16 Gold misses corrected from Form errors is shown in (9b), derived from the NNS response in (9a). Here, while (9b) is an appropriate form and triple, the triple is not found in the gold standard, because every NS respondent described the PDT item as a *raking* action, not a sweeping action.

(9) a. NNS: a men is swapping the leaves. ⇒ swap(man,leaf)
b. LM: a man is sweeping the leaves. ⇒ sweep(man,leaf)

Another example of a NNS Form error changed to an LM Gold miss under the Joint(NNS) experiment is the correction of *draw(artiest,portrait)* (seen above in (3)) to *draw(artist,portrait)*.

7 Summary and Outlook

We have implemented a system for automatically correcting NNS responses for visual stimuli by relying on a small set of known appropriate responses to influence the correction process. Even with a very limited gold standard, these corrections boosted coverage by 13.4% and decreased the total rate of Form and Triple errors by 13.7% (with potential for a decrease of 25.9%, as in the oracle experiments). These results can help guide the development of systems that aim to process the meaning of NNS statements, which contain a significantly higher rate of spelling errors compared to NS statements. There is much to be gained with a small amount of computational effort; as demonstrated here, more work needs to go into delineating a proper set of appropriate responses.

Indeed, we see the construction of a robust set of appropriate responses as the most immediate means of improving system performance. As NSs were shown to converge on a limited vocabulary for some items, while NNSs do not, simply collecting more NS responses would result in diminishing returns. Future work will need to uncover the best means of obtaining a sufficient set of responses to describe a picture, whether it involves a more sophisticated and

in-depth elicitation of NS responses or a deliberate attempt by the researchers at exhaustively describing the images. Moreover, as this work will ideally lead toward a game or ILT, it may be preferable to allow for "partial credit" (and the presentation of feedback) in the case of triples that do not constitute a complete match but may match one or two of the subject, verb, and object.

Similarly, as the correction module relies on the words used by NSs to influence corrections, expanding the list of "influential" words is likely to be beneficial. While in the current study this consisted of a simple list derived from the same responses in the gold standard, this is simply in keeping with (King and Dickinson, 2013) and may not be optimal. A more sophisticated approach could allow this influence to be probabilistic rather than binary, and could rely on methods like TF-IDF to determine which words in NS responses are particularly relevant to the item, and which words are incidental.

Another obvious source of improvement for future work is in the choice of training texts for the LM, which was shown to have serious biases against the contents of the PDT responses, which tend to describe physical actions or scenarios not common in newspaper text. Finding training texts that contain the necessary kinds of sentences but also the sheer volume needed to cover the variability of NNS responses is a challenge for future experiments in this area.

Given that this study primarily investigated transitive verbs, research on this problem will need to examine interactions with other types of constructions, including the definition of more elaborate semantic forms (Hahn and Meurers, 2012). Moving to a wider range of sentence types may require the use of a semantic role labeler or similar tools and has the potential to increase the complexity of spelling correction, due to, e.g., longer sentences.

Acknowledgments

We would like to thank the task participants, David Stringer for assistance in developing the task, and Kathleen Bardovi-Harlig, Marlin Howard and Jayson Deese for help in recruiting participants. We also thank Abigail Elston and Alex Rudnick for their helpful advice during the system development. Finally, for their insightful feedback, we would like to thank the two anonymous reviewers and the attendees of the computational linguistics colloquium series at Indiana University.

References

Atkinson, K. (1998). GNU Aspell. `http://aspell.net`.

Clarkson, P. and Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge Toolkit. In *Eurospeech*, volume 97, pages 2707–2710.

Dale, R., Anisimoff, I., and Narroway, G. (2012). HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62, Montréal.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC 2006*, Genoa, Italy.

DeSmedt, W. (1995). Herr Kommissar: An ICALL conversation simulator for intermediate German. In Holland, V. M., Kaplan, J., and Sams, M., editors, *Intelligent Language Tutors: Theory Shaping Technology*, pages 153–174. Lawrence Erlbaum, Mahwah, NJ.

Ellis, R. (2000). Task-based research and language pedagogy. *Language Teaching Research*, 4(3):193–220.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, MA.

Flor, M. (2012). Four types of context for automatic spelling correction. *TAL*, 53(2):61–99.

Flor, M. and Futagi, Y. (2012). On using context for automatic correction of non-word misspellings in student essays. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 105–115. Association for Computational Linguistics.

Flor, M., Futagi, Y., Lopez, M., and Mulholland, M. (2013). Patterns of misspellings in L2 and L1 English: A view from the ETS Spelling Corpus. In *Proceedings of the Second Learner Corpus Research Conference (LCR 2013)*.

Forbes-McKay, K. and Venneri, A. (2005). Detecting subtle spontaneous language decline in early Alzheimer's disease with a picture description task. *Neurological Sciences*, 26(4):243–254.

Graff, D., Kong, J., Chen, K., and Maeda, K. (2007). English Gigaword, Third Edition.

Hahn, M. and Meurers, D. (2012). Evaluating the meaning of answers to reading comprehension questions: A semantics-based approach. In *Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications (BEA7)*, pages 326–336, Montreal, Canada. Association for Computational Linguistics.

Heift, T. and Schulze, M. (2007). *Errors and Intelligence in Computer-Assisted Language Learning: Parsers and Pedagogues*. Routledge.

Hovermale, D. (2008). SCALE: Spelling Correction Adapted for Learners of English. Pre-CALICO Workshop on "Automatic Analysis of Learner Language: Bridging Foreign Language Teaching Needs and NLP Possibilities". March 18-19, 2008. San Francisco, CA.

Hovermale, D. (2010). An analysis of the spelling errors of L2 English learners. In *CALICO 2010 Conference, Amherst, MA, USA*.

King, L. and Dickinson, M. (2013). Shallow semantic analysis of interactive learner sentences. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–21, Atlanta, Georgia.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of ACL-03*, Sapporo, Japan.

Lachowicz, D. (2003). Enchant. `http://abisource.com/projects/enchant`.

Leacock, C., Chodorow, M., Gamon, M., and Tetreault, J. (2010). *Automated Grammatical Error Detection for Language Learners*. Synthesis Lectures on Human Language Technologies. Morgan Claypool.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Meurers, D. (2012). Natural language processing and language learning. In Chapelle, C. A., editor, *Encyclopedia of Applied Linguistics*. Blackwell.

Meurers, D., Ziai, R., Ott, N., and Bailey, S. (2011). Integrating parallel analysis modules to evaluate the meaning of answers to reading comprehension questions. *Special Issue on Free-text Automatic Evaluation. International Journal of Continuing Engineering Education and Life-Long Learning (IJCEELL)*, 21(4):355–369.

Petersen, K. A. (2010). *Implicit Corrective Feedback in Computer-Guided Interaction: Does Mode Matter?* PhD thesis, Georgetown University, Washington, DC.

Somasundaran, S. and Chodorow, M. (2014). Automated measures of specific vocabulary knowledge from constructed responses ('Use these words to write a sentence based on this picture'). In *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–11, Baltimore, Maryland.

Sproat, R., Black, A. W., Chen, S., Kumar, S., Ostendorf, M., and Richards, C. (2001). Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333.