A Appendix: Perplexity

The perplexity in the paper is formulated as follows:

$$PP(S) = (\prod_{s \in S} P(w_{1:t} = s))^{\frac{-1}{|S|}} \quad (1)$$

By definition, the perplexity of a model q on a test suit S is defined as follows:

$$PP(S) = 2^{H(p,q)} \quad (2)$$

where H is cross entropy, and p is the likelihood of each possible sample in the test suit. The definition of cross entropy is as follows:

$$H(p,q) = -\sum_{x \in S} p(X = x) log_2(q(X = x)) \quad (3)$$

where X is a random variable, and x is a possible value of the random variable. In a forward generative language model, the random variable is conditioned on the previous words. With test suite being a sequence of words $S = w_{1:T}$, the likelihood of each word in the sequence is $p(w_t) = \frac{1}{T}$, and the cross entropy of the model on the samples is:

$$H(p,q) = -\sum_{t=1}^{T} p(w_t) log_2(q(w_t|w_{1:t-1})) \quad (4)$$

$$= -\frac{1}{T} \sum_{t=1}^{T} log_2(q(w_t|w_{1:t-1})) \quad (5)$$

where w_t is a token at a time t, in a sequence with maximum T tokens, $w_{1:t} = w_1, w_2, ..., w_t$. Therefore the perplexity is:

$$PP(S) = 2^{-\frac{1}{T} \sum_{t=1}^{T} log_2(q(w_t|w_{1:t-1}))} \quad (6)$$

$$= (\prod_{t=1}^{T} q(w_t|w_{1:t-1}))^{-\frac{1}{T}} \quad (7)$$

Equation 7 is often used as definition of perplexity in language models (Goodman, 2001) and Equation 6 is its numeric computation to avoid underflow due to adding logits.

There are two ways to extend the definition to the case when perplexity is calculated for a collection of sentences. (i) We can treat the corpus as a long sequence of tokens and use the previous equations. (ii) We can use Equation 3 with a change to the model definition, from a token model to a sentence model. The benefit of this method is that it assigns the same likelihood for each sentence regardless of its length. In this case, the chain rule is used for the sentence model. The likelihood of each sentence is one over the number of sentences in the test suite, $p(s) = \frac{1}{|S|}$:

$$H(p,q) = -\sum_{s \in S} p(s) log_2(\hat{P}(s)) \quad (8)$$

$$= -\frac{1}{|S|} \sum_{s \in S} log_2(\hat{P}(s)) \quad (9)$$

Based on the chain rule, the sentence model can be calculated as follows:

$$\hat{P}(w_{1:T} = s) = \prod_{t=1}^{T} q(w_t|w_{1:t-1}) \quad (10)$$

$$log_2(\hat{P}(w_{1:T} = s)) = \sum_{t=1}^{T} log_2(q(w_t|w_{1:t-1})) \quad (11)$$

Perplexity in this case is defined as in Equation 1 here repeated as Equation12:

$$PP(S) = (\prod_{s \in S} \hat{P}(w_{1:T_s} = s))^{\frac{-1}{|S|}} \quad (12)$$

which instead of using the product is computed as a sum of logits from Equation 8 and 11.

B Appendix: Examples of images from Visual Genome

Figure 1 and Figure 2 are the examples from VisualGenome which their region descriptions are used in the paper as examples of relation-context substitution table.

C Appendix: Complete P-vectors

Figure 3 is the full presentation of P-vectors.

D Appendix: Similarity Judgment Dataset

In total 66 worker in Amazon Mechanical Turk annotated the word similarity. For each word pair, we collected 10 judgments. The word pairs vertically in random order were presented to annotators to judge their similarity. The input form was a slider in the web interface which they could freely adjust the indicator position between dissimilar and similar rating (Figure 4). In order to identify the bad annotators, we randomly asked the annotators to judge similarity between "*green*" and one of the spatial relations, we also asked similarity

Figure 1: image_id = 2367586
tall building above the bridge
bench below the green trees
car next to the water

Figure 2: image_id = 2320485
scissors above the pen
the pen is below scissors
a ball-pen next to the scissorts

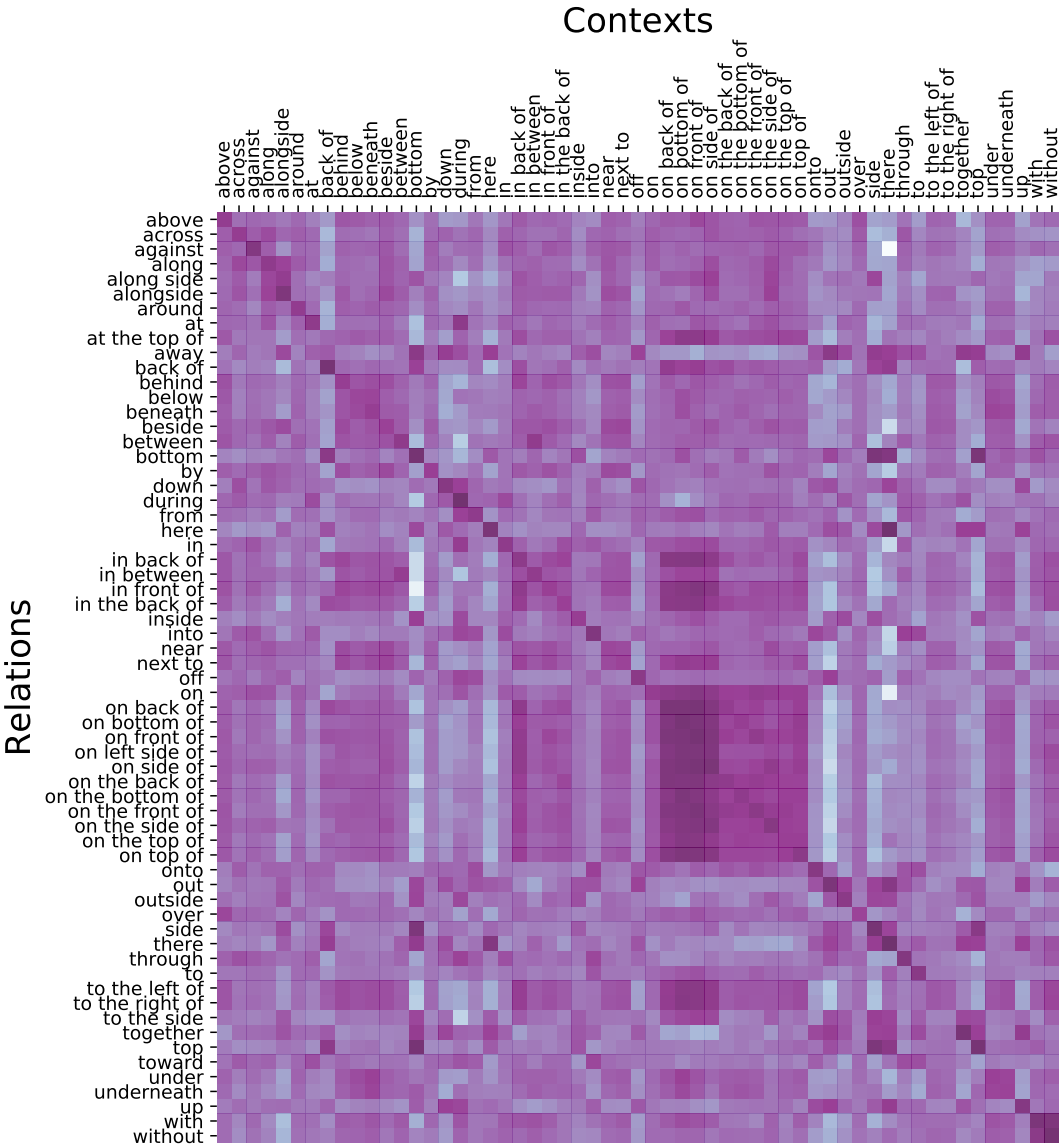

Figure 3: Perplexity vectors for 67 spatial relations on 57 context bins.

judgment between a spatial relation and itself. If the answer to similarity with green was higher than %60, or the answer for self similarity was lower than %90, all contributions of that worker were taken out from the dataset. This cleaning technique removed 9 workers in total, which left us about 7 annotation on each word pair.

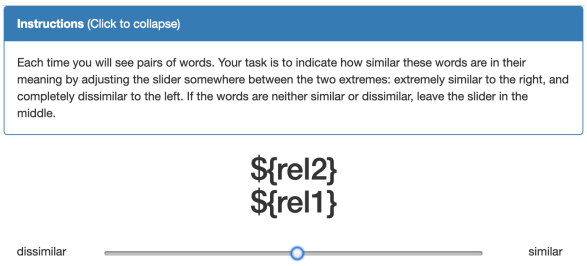

Figure 4: The layout which presents the similarity judgment question.

References

Joshua T Goodman. 2001. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434.