Supplemental Material: Learning Scalar Adjective Intensity from Paraphrases

Anonymous EMNLP submission

1 Constructing a Graph of Adjectival Paraphrases

Here we present further detail on the process of constructing JJGRAPH for reproducibility.

1.1 Dissecting Paraphrase Pairs

Building a graph to represent paraphrase pairs requires identification of paraphrases with the form (*RB* $JJ_u \leftrightarrow JJ_v$) in PPDB-XXL. Paraphrases in PPDB consist of phrase pairs, where each phrase can be one or more words long. Each pair is labeled with a syntactic part of speech; for this work we consider only adjectival phrase (ADJP) paraphrases.

Given a paraphrase pair, we denote as P_1 the phrase with longer token length, and P_2 the shorter phrase. We assume that P_2 consists of a single adjective, and P_1 consists of an adjective modified by an adverb. More specifically, within P_1 of length n, we identify the adjective as the last token, and the adverbial modifier the concatenated tokens from the first to $(n-1)^{th}$ token. For the purposes of this study, phrases where the adverb meets one of the following criteria are ignored: longer than 4 tokens; consists of a single character; consists of the word *not*; ends with one of the tokens *about*, *and*, *in*, *or*, *the*, or *to*; or contains digits.

1.2 Bootstrapping Intensifying Adverbs

Recall that building JJGRAPH requires a set of intensifying adverbs in order to select paraphrases of the form (*RB* $JJ_u \leftrightarrow JJ_v$) for inclusion in the graph. We used an iterative bootstrapping process to identify intensifying adverbs as outlined in Section 3.1 of the paper.

The process begins with a small seed set of adjective pairs with a known intensity relationship. For the seeds, we identified adjectival phrase paraphrases (P_1,P_2) in PPDB where the adjective in P_1 is in its base form (e.g., *hard*), and P_2 is either the superlative or comparative form of the same adjective (e.g., *harder*). Such pairs were identified by lemmatizing with NLTK's `WordNetLemmatizer` (Loper and Bird (2002)).

By definition, the base form of an adjective is less semantically intense than both its comparative and superlative forms (e.g., *hard* < *harder* < *hardest*). Thus, the adverb that precedes the base form of the adjective in P_1 is presumed to be an intensifying adverb. We add all adverbs identified in Round 1 of the process to an initial adverb list, R_1.

Next, we found additional paraphrases where the adverb in P_1 appears in R_1, but the adjectives in P_1 and P_2 could be anything. For example, where in Round 1 we identified *very* and *pretty* as intensifying adverbs, we found in Round 2 that *pleasant* and *delightful* and that *simple* and *plain* were also related by *very* and *pretty*, respectively. That is, *pleasant* < *delightful* and *simple* < *plain*.

Finally, in Round 3, we identified an additional set of intensifying adverbs by finding paraphrases with the new adjective pairs identified in Round 2. For example, in Round 2 we found the patterns *[adverb] pleasant* ↔ *delightful* and *[adverb] simple* ↔ *plain*, so in Round 3 we found all other paraphrases that fit those patterns (e.g., *more pleasant* ↔ *delightful*, *quite simple* ↔ *plain*). The adverbs in these phrases (e.g., *more*, *quite*) are intensifying. We add all such adverbs identified in Round 3 to an adverb set R_3, and take as our final set of intensifying adverbs $R = R_1 \cup R_3$.

In total, we identified 610 intensifying adverbs using this process. We also carried out a parallel process to identify de-intensifying adverbs, but found that the list of intensifiers was larger, and that the list of de-intensifying adverbs overlapped heavily with the list of intensifying adverbs. Thus we focused primarily on the intensifying adverbs for creating JJGraph.

Finally, JJGraph is formed by extracting all adjectival phrase paraphrases in PPDB of the form (*RB* $JJ_u \leftrightarrow JJ_v$), where the adverb belongs to the set of intensifiers. Each paraphrase is then represented by two nodes (JJ_u, JJ_v) and the directed adverb edge with label *RB* from JJ_u to JJ_v.

2 Dataset Generation

In this paper we utilized two previously-released datasets of gold standard adjective intensity rankings (de Melo and Bansal, 2013; Wilkinson and Oates, 2016), and also generated a third, new set of gold standard adjective scales through crowdsourcing. This section details the process of modifying the Wilkinson dataset from full- to half-scales for use in this study, and the process of creating the new crowdsourced dataset.

2.1 Adapting the Wilkinson Dataset

The Wilkinson dataset (Wilkinson and Oates, 2016) as published provides 12 full adjective scales between polar opposites, e.g. (*ancient*, *old*, *fresh*, *new*). We manually subdivided each scale into half scales for compatibility with the other datasets in this study, producing 21 half scales total. The procedure for dividing a full- into a half-scale was as follows:

1. If the full scale contains two central adjectives where the polarity shifts from negative to positive, sub-divide the scale between them (e.g. divide the scale (*simple*, *easy*, *hard*, *difficult*) between central adjectives *easy* and *hard*).

2. Otherwise, if the full scale contains a central neutral adjective, subdivide the full scale into halves with the neutral adjective belonging to both half scales (e.g. divide (*freezing*, *cold*, *warm*, *hot*) into (*freezing*, *cold*, *warm*) and (*warm*, *hot*)).

3. If any of the resulting half scales has length 1, delete it.

Table 1 enumerates the half-scales we generated from the full Wilkinson dataset.

hideous ugly || pretty beautiful gorgeous
dark dim || light bright
same alike similar || ~~different~~
simple easy || hard difficult
parched arid dry || damp moist wet
|| few some several many
horrible terrible awful bad || good great wonderful awesome
freezing cold warm || warm hot
ancient old || fresh new
~~slow~~ || quick fast speedy
miniscule tiny small || big large huge enormous gigantic
idiotic stupid dumb || smart intelligent

Table 1: Converting the 12 Wilkinson full scales to 21 half scales. The || symbol denotes the location where full scales are split into half scales. Strike-through text indicates a half-scale was deleted due to having a single adjective.

2.2 Building the Crowd Dataset

In order to maximize coverage of our JJGraph vocabulary, we also generated a new dataset of adjective intensity half-scales. Our general approach was, first, to compile *clusters* of adjectives describing a single attribute, and second, to rank adjectives within each cluster by their intensity.

2.2.1 Generating Adjective Sets

We generated clusters of adjectives modifying a shared attribute by partitioning sets of related adjectives associated with a single target word in JJGraph. For example, given the target adjective *hot*, we might generate the following clusters from the set of associated words *warm, heated, boiling, attractive, nice-looking, new*, and *popular*:

$$
\begin{aligned}
c_1 &= \{\text{warm, heated, boiling}\} \\
c_2 &= \{\text{attractive, nice-looking}\} \\
c_3 &= \{\text{new, popular}\}
\end{aligned}
$$

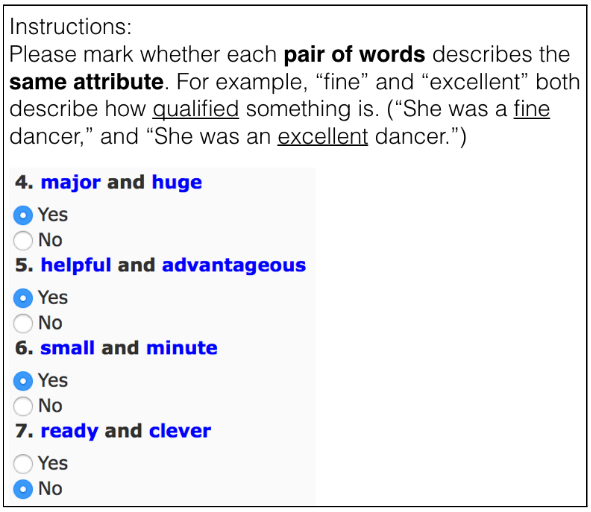
Each cluster represents a semantic sense of the target adjective, and thus the adjectives within a cluster can be ordered along a single scale of increasing intensity. Clusters do not need to be disjoint, as some adjectives have multiple senses.

Partitioning the sets was accomplished with the aid of crowd workers on Amazon Mechanical Turk (MTurk) in two stages. Here we describe the process.

We began by selecting target adjectives with high centrality in JJGraph around which to create gold standard clusters. An adjective has "high centrality" if it is among the 200 most central nodes according to two of three centrality measurements – betweenness centrality, closeness centrality, and degree centrality. With this criterion, we selected 145 target adjectives from JJGRAPH around which adjective sets were generated.

For each target adjective, we then generated a candidate set of related adjectives to pass to our first MTurk task, which asked workers to remove unrelated adjectives from the candidate sets. We compile an initial candidate set for each of the 145 target adjectives by collecting the first 20 words encountered in a breadth-first search starting at the adjective in JJGRAPH.

Our first MTurk task aimed to remove unrelated adjectives from the 145 candidate sets (see Figure 1). We presented workers with pairs of adjectives, one being the target adjective and the other a word from that target's candidate set. Three Turkers assessed each pair of adjectives. If a majority of Turkers declared that a pair of adjectives did not describe the same attribute, then the candidate word was removed from that target's set.



Instructions:
Please mark whether each **pair of words** describes the **same attribute**. For example, "fine" and "excellent" both describe how qualified something is. ("She was a fine dancer," and "She was an excellent dancer.")

4. major and huge
● Yes
○ No

5. helpful and advantageous
● Yes
○ No

6. small and minute
● Yes
○ No

7. ready and clever
○ Yes
● No

Figure 1: First MTurk HIT for constructing gold standard adjective clusters. Each question consists of a target adjective (left) and a cluster candidate adjective (right).

Once we had a clean set of related adjectives for each target, our second task asked workers to partition the related words (Figure 2). Between 2 and 10 Turkers constructed a clustering for each target adjective. Once a predefined level of agreement was reached among Turkers for a target adjective's clusters, the clusters were deemed "gold."

In total, we constructed gold standard clusterings for 145 adjectives. Each candidate set was partitioned into an average of 3.26 clusters.

2.2.2 Ranking Adjectives in a Cluster

Given a clustering of related adjectives for each of the 145 target words, our next step was to ask MTurk workers to order adjectives within a single cluster by intensity.

We completed the ordering in a pairwise fashion. For each adjective cluster, we asked 3 MTurk workers to evaluate – for each pair of adjectives (j_u, j_v), whether j_u was less, equally, or more intense than j_v. The inter-annotator agreement on this task (Cohen's kappa) was $\kappa = 0.53$, indicating moderate agreement.

Finally, we filtered each cluster to include only adjectives with a unanimous, consistent global ranking. More specifically, if a cluster has adjectives j_u, j_v, and j_w, and workers unanimously agree that $j_u < j_v$ and $j_v < j_w$, then workers must also unanimously agree that $j_u < j_w$ for the ranking to be consistent. After this final step, our dataset consisted of 79 remaining clusters having from 2 to 8 ranked adjectives each (mean 3.18 adjectives per cluster).

3 Implementation Details for Global Ranking Model

We adopt the mixed-integer linear programming (MILP) approach of de Melo and Bansal (2013) for generating a global intensity ranking. This model takes a set of adjectives $A = \{a_1, \ldots, a_n\}$ and directed, pairwise adjective intensity scores $score(a_i, a_j)$ as input, and assigns each adjective a_i a place along a linear scale $x_i \in [0, 1]$. The adjectives' assigned values along the scale define the global ordering. Because the predicted weights used as input may be inconsistent, containing cycles, the model must resolve these by choosing the globally optimal solution.

Recall that all pairwise scoring metrics in this study produce a positive score for adjective pair $(j_u,\ j_v)$ when it is likely that $j_u < j_v$, and a negative score when $j_u > j_v$. Consequently, the MILP approach should result in $x_u < x_v$ when $score(j_u, j_v)$ is positive, and $x_u > x_v$ otherwise. This goal is achieved by maximizing the objective function:

$$\sum_{u,v} \text{sign}(x_v - x_u) \cdot score(j_u, j_v) \qquad (1)$$

de Melo and Bansal (2013) propose the following MILP formulation for maximizing this objec-

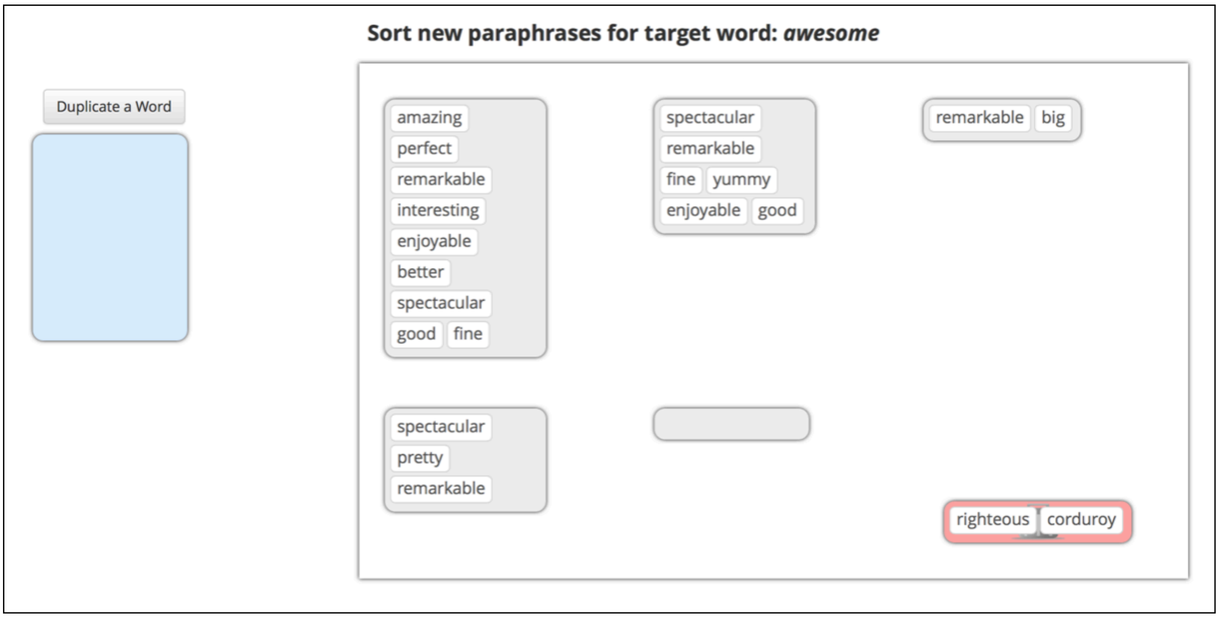


Figure 2: Second MTurk HIT for constructing gold standard adjective clusters.

tive:

$$\begin{array}{lrl}
\max\limits_{u,v} & \sum\limits_{u,v}(w_{uv} - s_{uv}) \cdot score(j_u, j_v) & \\
\text{s.t.} & d_{uv} = x_v - x_u & \forall u,v \in N \\
& d_{uv} - w_{uv}C \leq 0 & \forall u,v \in N \\
& d_{uv} + (1 - w_{uv})C > 0 & \forall u,v \in N \\
& d_{uv} + s_{uv}C \geq 0 & \forall u,v \in N \\
& d_{uv} - (1 - s_{uv})C < 0 & \forall u,v \in N \\
& x_u \in [0,1] & \forall u \in N \\
& w_{uv} \in \{0,1\} & \forall u,v \in N \\
& s_{uv} \in \{0,1\} & \forall u,v \in N
\end{array} \tag{2}$$

The variable d_{uv} is a difference variable that captures the difference between x_v and x_u. The constant C is an arbitrarily large number that is at least $\sum_{u,v} |score(j_u, j_v)|$. The variables w_{uv} and s_{uv} are binary indicators that correspond to a *weak-strong* or *strong-weak* relationship between j_u and j_v respectively; the objective encourages $w_{uv} = 1$ when $score(j_u, j_v) > 0$, and $s_{uv} = 1$ when $score(j_u, j_v) < 0$. While de Melo and Bansal (2013) also propose an additional term in the objective that incorporates synonymy information, we do not implement this part of the model.

4 Full Results

Only the best results for combined scoring methods were given in the main body of the paper. Here we provide the full results for all combinations attempted on both experiments.

References

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Philadelphia, Pennsylvania.

Gerard de Melo and Mohit Bansal. 2013. Good, great, excellent: Global inference of semantic intensities. *Transactions of the Association for Computational Linguistics*, 1:279–290.

Bryan Wilkinson and Tim Oates. 2016. A gold standard for scalar adjectives. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, Portoro, Slovenia.

Method	%OOV	Acc.	P	R	F
$score_{\text{socal+pat+pp}}$	0.06	0.642	0.684	0.683	0.684
$score_{\text{socal+pp+pat}}$	0.06	0.642	0.678	0.676	0.677
$score_{\text{socal+pp}}$	0.09	0.634	0.690	0.663	0.676
$score_{\text{socal+pat}}$	0.07	0.634	0.680	0.670	0.675
deMarneffe (2010)	0.02	0.610	0.597	0.594	0.596
$score_{\text{socal}}$	0.26	0.504	0.710	0.481	0.574
$score_{\text{pp+pat}}$	0.06	0.504	0.559	0.547	0.553
$score_{\text{pp+pat+socal}}$	0.06	0.504	0.559	0.547	0.553
$score_{\text{pp+socal+pat}}$	0.06	0.504	0.559	0.547	0.553
$score_{\text{pp+socal}}$	0.09	0.496	0.568	0.533	0.550
$score_{\text{pp}}$	0.09	0.496	0.568	0.533	0.550
$score_{\text{pat+pp}}$	0.06	0.423	0.532	0.517	0.524
$score_{\text{pat+socal+pp}}$	0.06	0.423	0.532	0.517	0.524
$score_{\text{pat+pp+socal}}$	0.06	0.423	0.532	0.517	0.524
$score_{\text{pat+socal}}$	0.07	0.415	0.528	0.504	0.516
$score_{\text{pat}}$	0.07	0.407	0.524	0.491	0.507
all-“YES”	0.00	0.691	0.346	0.500	0.409

Table 2: **Full IQAP Results.** Accuracy and macro-averaged precision (P), recall (R), and F1-score (F) over *yes* and *no* responses on 123 question-answer pairs. The percent of pairs having one or both adjectives out of the score vocabulary is listed as %OOV. Rows are sorted by descending F1-score.

		Score Accuracy (before ranking)		Global Ranking Results		
Test Set	Score Type	Coverage	Pairwise Acc.	Pairwise Acc.	Avg. τ_b	ρ
deMelo	$score_{\text{pat}}$	0.480	0.844	0.650	0.633	0.583
	$score_{\text{pp}}$	0.325	0.458	0.307	0.071	0.09
	$score_{\text{socal}}$	0.277	0.546	0.246	0.110	0.019
	$score_{\text{pat+pp}}$	0.623	0.742	0.619	0.543	0.511
	$score_{\text{socal+pp}}$	0.478	0.523	0.380	0.162	0.106
	$score_{\text{pat+socal}}$	0.609	0.757	0.653	0.609	0.533
	$score_{\text{pat+pp+socal}}$	0.698	0.718	0.637	0.537	0.463
	$score_{\text{pat+socal+pp}}$	0.698	0.722	0.644	0.564	0.482
	$score_{\text{pp+socal+pat}}$	0.698	0.635	0.579	0.393	0.327
	$score_{\text{pp+pat+socal}}$	0.698	0.661	0.599	0.437	0.372
	$score_{\text{socal+pp+pat}}$	0.698	0.647	0.589	0.430	0.341
	$score_{\text{socal+pat+pp}}$	0.698	0.680	0.613	0.496	0.395
Crowd	$score_{\text{pat}}$	0.112	0.784	0.321	0.203	0.221
	$score_{\text{pp}}$	0.738	0.676	0.597	0.437	0.405
	$score_{\text{socal}}$	0.348	0.757	0.421	0.342	0.293
	$score_{\text{pat+pp}}$	0.747	0.696	0.627	0.481	0.432
	$score_{\text{socal+pp}}$	0.812	0.687	0.621	0.470	0.465
	$score_{\text{pat+socal}}$	0.412	0.750	0.476	0.373	0.298
	$score_{\text{pat+pp+socal}}$	0.821	0.686	0.630	0.462	0.440
	$score_{\text{pat+socal+pp}}$	0.821	0.686	0.624	0.465	0.472
	$score_{\text{pp+socal+pat}}$	0.821	0.670	0.630	0.456	0.435
	$score_{\text{pp+pat+socal}}$	0.821	0.670	0.630	0.456	0.435
	$score_{\text{socal+pp+pat}}$	0.821	0.690	0.633	0.481	0.480
	$score_{\text{socal+pat+pp}}$	0.821	0.694	0.639	0.495	0.480
Wilkinson	$score_{\text{pat}}$	0.443	0.852	0.475	0.441	0.435
	$score_{\text{pp}}$	0.795	0.753	0.639	0.419	0.450
	$score_{\text{socal}}$	0.311	0.895	0.312	0.317	0.422
	$score_{\text{pat+pp}}$	0.885	0.833	0.738	0.605	0.564
	$score_{\text{socal+pp}}$	0.795	0.773	0.672	0.484	0.565
	$score_{\text{pat+socal}}$	0.639	0.846	0.59	0.503	0.506
	$score_{\text{pat+pp+socal}}$	0.885	0.833	0.738	0.605	0.564
	$score_{\text{pat+socal+pp}}$	0.885	0.833	0.754	0.638	0.600
	$score_{\text{pp+socal+pat}}$	0.885	0.750	0.672	0.426	0.414
	$score_{\text{pp+pat+socal}}$	0.885	0.750	0.672	0.426	0.414
	$score_{\text{socal+pp+pat}}$	0.885	0.769	0.705	0.492	0.504
	$score_{\text{socal+pat+pp}}$	0.885	0.833	0.754	0.638	0.611

Table 3: **Full Global Ranking Results.** Pairwise relation prediction and global ranking results for each score type in isolation, and for all combinations of 2 and 3 score types attempted on each dataset.