Ranking Scalar Adjectives: A Comparison of Different Transformers Encoder Models

Alon Mizrahi, Itamar Levtzur, Noam Cohen

September 22, 2023

Abstract

Scalar adjectives describe a property of a noun at different degrees of intensity. The ability to rank scalar adjectives of a given scale (i.e. recognizing the increasing intensity between pretty, beautiful and gorgeous) implies a deep understanding of language, whether the ranker is a human or a machine learning model. We apply two different ranking methods on contextualized representations of adjectives of the same scale (e.g. *warm, hot, boiling*) to evaluate an intensity grade for each adjective, and compare the results with ground truth data. We proceed to run this procedure using open-source state-of-the-art encoder-only models. Our results show which model performs best at detecting intensity levels of scalar adjectives, indicating a richer knowledge of language in this aspect.

1 Introduction

Encoders are a key component of the transformers encoder-decoder architecture [Vaswani et al., 2017]. They are responsible for analyzing the input sequence and representing it in a way to decoder can understand. In NLP, transformer encoders encode sequences of tokens into rich, contextualized representations (embeddings) that are then fed to a decoder to perform sequence to sequence tasks, such as machine translation or text summarization. In addition, pretrained encoders can be fine tuned on specific tasks by attaching a head (e.g. single-layer perceptron) on top of them. In this paper, our focus is solely on the embeddings produced by these encoders, without making use of a decoder or head. We compare five encoder-only models on the task of ranking scalar adjectives: BERT [Devlin et al., 2019], DistilBERT [Sanh et al., 2020], ALBERT [Lan et al., 2020], RoBERTa [Liu et al., 2019] and ELECTRA [Clark et al., 2020]. To compare adjectives of some scale s, we pass $|s|$ instances of the same sentence to the encoder with a different adjective $a \in s$ present in each instance. Then, we extract the adjective representations of all instances and compare them by using two methods which we describe later. In each method, we calculate an intensity score for every $a \in s$, which we then compare to ground truth data to evaluate a final score for the encoder model we used. The full implementation of the methods discussed in this paper is available on github.

2 Related Work

The analysis of scalar adjectives can be split into two tasks: grouping together adjectives of the same scale, and ranking them by their intensities. Standard clustering algorithms were used by [Hatzivassiloglou and McKeown, 1993] to group semantically related adjectives.

For the second task, [de Melo and Bansal, 2013] used a pattern-based approach for ranking scalar adjectives (e.g. we can infer from *"not only* $\boldsymbol{x}$*, but* $\boldsymbol{y}$*"* that $x < y$). [Soler and Apidianaki, 2020] defined the ranking methods used here to evaluate how well BERT ranks scalar adjectives on adjective datasets, and on an Indirect Question Answering task.

[Mikolov et al., 2013] show how word representations can be used to capture syntactic and semantic

Adjective Dataset	Scale Name	Adjectives
Crowd	remarkable	fine, remarkable, impressive
	tiny	little, tiny
Wilkinson	large	big, large, huge, enormous, gigantic
	good	good, great, wonderful, awesome

Table 1: Adjective scales from Crowd and Wilkinson datasets

regularities, and [Kim and de Marneffe, 2013] take one step further to derive adjectival scales from these representation. A similar approach is used here as the second ranking method.

3 Data

3.1 Adjectives

We experiment with three scalar adjective datasets based on adjective intensity estimates:

- DeMelo [de Melo and Bansal, 2013] - 87 half-scales.
- Crowd [Cocos et al., 2018] - 79 half-scales.
- Wilkinson [Wilkinson, 2017] - 21 half-scales.

Two of the original datasets contain full scales (e.g. *freezing* to *boiling*). To avoid changing the meaning of a sentence when substituting adjectives from the same scale, we use a transformed version of these datasets that breaks down full scales to half-scales (*freezing* to *cold*, *warm* to *boiling*). This also prevents us from generating illogical sentences like "Icy winds make winter days feel warm" by substituting the original adjective *cold* with *warm* from the same scale. For the remainder of the paper, we use the term "scale" to refer to "half-scales". See Table 1 for examples of adjective scales.

3.2 Sentences

We extract sentences from two types of corpora - reviews (movie, business reviews) and news items. We then evaluate adjective rankings separately for each corpus. The reason for this separate evaluation is that sentences sourced from internet reviews and news items differ by nature and tone. Internet reviews often contain personal opinions conveyed in an informal manner (*"Horrible movie, it looked so* ***cheap*** *too"*), whereas news items focus on objective reporting in a more formal manner (*"Politicians approved* ***cheap*** *power for well-connected corporations"*). As the results later show, evaluation scores may differ on these two types of corpora.
We used datasets to gather examples for both reviews and news sequences.

- For reviews sequences, we used imdb, rotten tomatoes and yelp reviews.
- For news items, we used cc_news, a dataset derived from Common Crawl.

We took 50k entries (sequences) from each corpus type.

4 Sentence Collection

For each adjective dataset D, corpus C, scale $s \in D$ and adjective $a \in s$, we parse all sentences in C and add sentences $t \in C$ that contain the adjective a to a set t_a. This constructs a set of sentences t_a that contain a for every $a \in s$. We set an upper bound of at most 100 sentences per adjective. See Table 2 for examples.

Adjective Dataset	Corpus	Scale	Adjective	Sentences
DeMelo	reviews	pretty-gorgeous	pretty	The actress is pretty. Pretty restaurant. ...
			beautiful	Beautiful scenes. Beautiful snowy mountains. ...
	news	pretty-gorgeous	pretty	Pretty weekend for tourists. ...
			beautiful	Beautiful species were found. ...

Table 2: Collected sentences for the adjective scale *pretty-beautiful-gorgeous* for each corpus type in DeMelo dataset

5 Data Cleaning

5.1 Scalar Adjectives

The original adjective datasets contain ties (e.g. *big, huge, immense | enormous* where *immense* is tied with *enormous*). The ranking methods we apply calculate an intensity score $int(a) \in [0, 1]$ for every scalar adjective a in scale s. This means that we had to come up with a diff value where two adjective $a_1, a_2 \in s$ are counted as a tie if $|int(a_1) - int(a_2)| <$ diff. We find choosing a value for diff rather arbitrary and inconsistent. For this reason, we decided to remove all ties from the datasets, taking only the first word when encountering a tie (*big, huge, immense | enormous* $\rightarrow$ *big, huge, immense*).

5.2 Sentences

We applied three cleaning methods sequentially on the sentences we parse.

5.2.1 Part-of-Speech Parsing

Many adjective words can serve as non-adjectives (e.g. *pretty* is used as an adverb in *"Pretty good movie"* and as an adjective in *"Pretty woman"*). For each sentence t that contains an adjective a, we used NLTK's part of speech parser to check whether the word a found in t is indeed used as an adjective, and add t to the set t_a only if it is.

5.2.2 Linguistic Acceptability

When substituting a scalar adjective in a sentence with a different adjective from the same scale, the sentence might not remain grammatically correct (*"The music had a **cool** beat"* $\rightarrow$ *"The music had a **freezing** beat"*). This might happen with scalar adjectives that belong to more than one scale. We fine-tuned BERT on the CoLA dataset [Warstadt et al., 2019] to classify whether a sentence is grammatically correct or not, achieving an accuracy of 85% on the evaluation set. Then, for every sentence that contains an adjective $a \in s$, we generate $|s| - 1$ new sentences of a substituted with the other adjectives $a' \in s, a' \neq a$, and pass them to the fine-tuned model. We only add the sentence t to t_a if the fine-tuned model classifies all $|s - 1|$ sentences as grammatically correct.

5.2.3 Context Deviation

Adjective substitution might keep the sentence grammatically correct but change its meaning (*"A **warm** person"* $\rightarrow$ *"A **hot** person"*). As we aim to capture only intensity changes of scalar adjectives, we'd like to avoid the noise of these context changes. For all the sentences that pass the two previous tests, we give a scale-similarity score for each sentence, indicating how well substituting adjectives from that scale preserved the meaning of the sentence. To do so, we passed the original sentence along with the sentences generated through substitution to BERT, and calculated the cosine similarity of the CLS representation of the original sentence with those of the other sentences, averaging the results. The

	Adjective Dataset	Corpus	*p-acc*	τ
Method 1	DeMelo	reviews	0.639	0.278
		news	0.633	0.266
	Crowd	reviews	0.701	0.403
		news	0.701	0.403
	Wilkinson	reviews	0.826	0.652
		news	0.783	0.565

Table 3: Evaluation scores on BERT using the first ranking method

CLS token is a special token used in BERT to capture a contextualized representation of the entire sequence. We then took the top 20 sentences by scale-similarity of all adjectives $a \in s$.

After collecting 20 sentences for each scale, we generate $|s| - 1$ new sentences from every sentence by substituting the original adjective $a \in s$ with all other adjectives in s. Every original sentence now has $|s|$ instances with a different adjective $a \in s$ in each instance, so there's a total of $20 \cdot |s|$ instances per scale.

6 Scalar Adjectives Ranking

We use two methods to give an intensity score for each adjective $a \in s$ based on its contextualized representation. Note that an adjective may be represented by more than one token, depending on the tokenizer used by each model. In that case we use the standard method of averaging all the output embeddings of these tokens to get a single embedding.

6.1 Ranking Relatively to the Extreme Adjective

The first ranking method that we used involves around comparing the representation of the extreme adjective in each scale (the one with the most intensity) $a_{ext} \in s$ with all the other adjectives $a \in s, a \neq a_{ext}$. An intensity score is given to an adjective based on the cosine similarity of its embedding and the embedding of a_{ext}, that is, $\forall a \in s, int(a) = \frac{emb(a) \cdot emb(a_{ext})}{||emb(a)|| \cdot ||emb(a_{ext})||}$. We ignore the intensity score of a_{ext} that always equals 1 and compare the intensities of all $a \in s, a \neq a_{ext}$. For this reason, scales with 2 adjectives are ignored when using this method. To compute the final intensity score for every adjective in some scale s, we apply this procedure 20 times (once for each original sentence), where in every run we send $|s|$ sentence instances to the embedding pipeline with a different adjective presented in each instance. After that, we average the cosine similarity results for each adjective $a \neq a_{ext}$ to achieve its final score. We order the adjectives by the intensity scores to get their ranking and compare it with the ground truth data in the scalar adjectives dataset. We evaluate the quality of the ranking using pairwise accuracy *p-acc* $\in [0, 1]$ and Kendall's correlation coefficient $\tau \in [-1, 1]$. See Table 3 for evaluation scores on all datasets using BERT as the encoder model.

	Adjective Dataset	Corpus	*p-acc*	τ
Method 2	DeMelo	reviews	0.767	0.534
		news	0.767	0.534
	Crowd	reviews	0.817	0.634
		news	0.805	0.610
	Wilkinson	reviews	0.875	0.750
		news	0.938	0.875

Table 4: Evaluation scores on BERT using the second ranking method

6.2 Ranking with a Global Intensity Vector

Our second adjective ranking method is derived from the well-known technique of constructing a "gender" vector by subtracting vectors of gender-specific words (e.g. $\overrightarrow{king} - \overrightarrow{queen}$, $\overrightarrow{actor} - \overrightarrow{actress}$) to construct a gender subspace of the embedding space spanned by these difference vectors. Similar to that, in this method we construct an "intensity" vector by subtracting the mild adjective a_{mild} in each scale from the extreme adjective a_{ext}, resulting in an adjective intensity vector $\overrightarrow{int} = \overrightarrow{a_{ext}} - \overrightarrow{a_{mild}}$. Then, we define the intensity of some adjective a by the cosine similarity of its embedding with $\overrightarrow{int}$. For every sentence in some scale s, we send its 2 instances of a_{mild} and a_{ext} to the encoder's pipeline and subtract their embeddings $emb(a_{ext}) - emb(a_{mild})$ to gain an intensity vector for that sentence. We average all 20 intensity vectors to get an intensity vector $\overrightarrow{(int_s)}$ for that scale. We then compute a single intensity vector for the adjective dataset $\overrightarrow{int_D}$ by averaging all the scale vectors. For neutral evaluation, we split every adjective dataset to train and test sets. We compute $\overrightarrow{int_D}$ using scales from the train set, and evaluate the rankings of adjectives from the test set. We use the same evaluation metrics from the previous method. See Table 4 for an evaluation of this method on BERT.

7 Evaluation on Different Encoder Models

Finally, we run both methods using the following encoder models: BERT, DistilBERT, ALBERT, RoBERTa and ELECTRA. We use the *base* version of each model. For ELECTRA, we use the Generator version that was trained on a task of token replacement, which fits better to the nature of this project. For the goal of comparing different models, we will be showing *p-acc* scores. Since we dropped tied adjectives from the original datasets, the model with the highest *p-acc* score will always have the highest τ score as well (although the values still differ). See Table 5 for evaluations on all models using both methods. We highlight the highest score for every dataset-corpus pair. We can see that Google's ELECTRA wins with the highest score on most datasets, and BERT being in second place. Also, there is a noticeable difference in the scores of each adjective dataset, so a case can be made for having our methods rank the datasets. DeMelo was constructed by applying pattern-based algorithms on text corpora. The other datasets were constructed (fully or partially) by crowdsourcing, resulting in adjective scales with better quality.

	Dataset	Corpus	Model				
			BERT	**DistilBERT**	**ALBERT**	**RoBERTa**	**ELECTRA**
Method 1	DeMelo	reviews	0.639	0.598	**0.651**	0.586	0.627
		news	0.633	0.592	**0.645**	0.550	0.604
	Crowd	reviews	0.701	0.623	0.636	0.571	**0.727**
		news	**0.701**	0.675	0.623	0.636	**0.701**
	Wilkinson	reviews	0.826	0.739	0.826	0.696	**0.913**
		news	0.783	0.783	0.913	0.609	**0.957**
Method 2	DeMelo	reviews	0.767	0.738	0.689	**0.796**	0.699
		news	**0.767**	0.748	0.699	**0.767**	0.709
	Crowd	reviews	0.817	0.780	0.793	0.817	**0.841**
		news	0.805	0.793	0.805	0.793	**0.829**
	Wilkinson	reviews	0.875	0.875	0.875	0.875	**0.938**
		news	**0.938**	**0.938**	0.875	0.875	**0.938**

Table 5: Pairwise accuracy scores of all encoder models using two ranking methods

8 Conclusion

We have shown how transformer encoders store rich information of intensity levels of scalar adjectives, which can then be used to rank adjectives from similar scales. We defined two methods for ranking scalar adjectives which we then use to compare SOTA encoder-only models. Furthermore, we could see that the second method used for ranking which constructs a global "intensity" vector performs better on all datasets. This strengthens the claim that word properties such as intensity levels of scalar

adjectives or genders can be extracted from a set of embedding vectors and used for different purposes. Our results show that ELECTRA and BERT perform best on most datasets. This can be leveraged for helping one decide which model they should use for a task where the degrees of intensity of scalar adjectives matter (e.g. multiclass text classification, where the intensity level of a scalar adjective in a sentence may be a major factor for the predicted class).

9 Future Work

For future work, we propose expanding the set of methods that we used for scalar adjectives ranking. For examples, the first method could easily be expanded to involve around a_{mild} when comparing scalar adjectives, meaning we measure the similarity of every adjective's embedding with the embedding of the adjective with the lowest intensity level in the scale a_{mild}, rather than a_{ext} which is what we currently do. A mixture of the two approaches may also be considered. Additionally, we could see much better results on scalar adjective scales that were constructed by crowdsourcing rather than auto generated scales, so perhaps putting effort on building such datasets from crowdsourcing may produce better and more coherent results for every model. Lastly, these methods can be used as-is for other languages besides English, such as Hebrew.

References

[Clark et al., 2020] Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators.

[Cocos et al., 2018] Cocos, A., Wharton, S., Pavlick, E., Apidianaki, M., and Callison-Burch, C. (2018). Learning scalar adjective intensity from paraphrases. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1752–1762, Brussels, Belgium. Association for Computational Linguistics.

[de Melo and Bansal, 2013] de Melo, G. and Bansal, M. (2013). Good, great, excellent: Global inference of semantic intensities. *Transactions of the Association for Computational Linguistics*, 1:279–290.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

[Hatzivassiloglou and McKeown, 1993] Hatzivassiloglou, V. and McKeown, K. R. (1993). Towards the automatic identification of adjectival scales: Clustering adjectives according to meaning. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, ACL '93, page 172–182, USA. Association for Computational Linguistics.

[Kim and de Marneffe, 2013] Kim, J.-K. and de Marneffe, M.-C. (2013). Deriving adjectival scales from continuous space word representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1625–1630, Seattle, Washington, USA. Association for Computational Linguistics.

[Lan et al., 2020] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations.

[Liu et al., 2019] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

[Sanh et al., 2020] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2020). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

[Soler and Apidianaki, 2020] Soler, A. G. and Apidianaki, M. (2020). Bert knows punta cana is not just beautiful, it's gorgeous: Ranking scalar adjectives with contextualised representations.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

[Warstadt et al., 2019] Warstadt, A., Singh, A., and Bowman, S. R. (2019). Neural network acceptability judgments.

[Wilkinson, 2017] Wilkinson, B. (2017). Identifying and ordering scalar adjectives using lexical substitution.