Revisiting Automatically-Generated Adjectival Scales with Continuous Space Word Representations

Gil Landau
glandau@seas.upenn.edu

Abstract

The goal of this study is to examine, replicate, and extend the model proposed by Kim and de Marneffe (2013). Their model uses the continuous space word representations described by Mikolov et al. (2010) to vectorize adjectives and discuss their relationships, with regards to their adjectival scale and relative intensity, in the new, high-dimensional, continuous space. The goal is discover what information and connections can be gleaned from these representations.

This study reviews and critiques a number of alternative approaches to generating an adjectival scale, and evaluates the performance of the original model on an expanded dataset and on the datasets of alternative models. The conclusion is clear: continuous space word representations are meaningful, but are inconsistent in determining adjectival scales.

1 Introduction

Continuous space word representations generated by neural networks capture syntactic and semantic meaning. The continuous model creates an n-dimensional space to represent a word, as compared to an n-gram model, which more directly bounds words to their discrete contexts. This makes them ideal to examine more complex relationships between words.

This paper attempts to use that meaning to construct a scale for adjective word representations. Cosine similarity is a good way to measure where a particular word fits on the scale (or what word fits at a particular point on the scale). Under the assumption of a linear relationship, one can use cosine similarity to narrow down the correct choice for a scale. For example, to find the comparative adjective, one can find the word closest to the middle of the superlative and base adjective. Similarly, one can determine which adjective best fits a scale, when given a number of options, by measuring their similarity to words on the scale.

Our model trains the word2vec model developed by Mikolov et al. (2013a), on the Google News data set (6B words, 3M word vectors with 300 dimensions). Our test set includes adjectival scales introduced by Wilkinson and Oates (2016) and de Melo and Bansal (2013), as well as a more expansive dataset generated using the intensity scales introduced by Taboada et al. (2011).

We generate both *full* and *half* scales using the datasets and test the performance of our model on both. The difference between a *full* adjectival scale and a *half* adjectival scale is a matter of extremes. We define a *full* adjectival scale as an adjectival scale that goes from antonym to antonym, centering around a neutral or transitioning adjective. By contrast, I define a *half* adjectival scale as an adjectival scale that only has increasing intensity, centering around a comparative adjective. So, for example, *hot*, *lukewarm*, *cold* versus *tepid*, *warmer*, *hot* are full scale and half scale, respectively.

We compare our approach and results to those of Wilkinson and Oates (2016) and de Melo and Bansal (2013). Notably, I do not use the question-answer approach used by Kim and de Marneffe (2013) nor their IQAP data-set to determine accuracy, opting instead for a more explicit generation of an adjectival scale.

2 Model and related work

This paper is based on the observations and experiments of Kim and de Marneffe (2013), which use the continuous word representations described by Mikolov et al. (2011) and expanded on in the recurrent neural network language model (RNNLM) discussed in Mikolov et al. (2013c). That paper trains the RNNLM on the Broadcast News dataset

(320M words) with dimensionality 1,600.

I use a slightly different approach, word2vec, described in Mikolov et al. (2013a) (specifically the skip-gram model) and trained on the Google News dataset (3M word vectors) with dimensionality 300.

To summarize the two models (from Mikolov et al. (2013a)):

First, continuous word vectors are learned using a simple model (will be explained later), and then the N-gram NNLM is trained on top of these word representations. The main difference between CBOW and skip-grams is that CBOW presents a word based on the surrounding contexts, and skip-grams presents contexts based on a word.

The probabilistic feedforward NNLM consists of input, projection, hidden layers, and output layers (though some models don't need any hidden layers!). At the input layer, N previous words are encoded using 1-of-V coding, where V is size of the vocabulary. The input layer is then projected to a projection layer P that has dimensionality N $\times$ D, using a shared projection matrix.

Continuous Bag of Words (CBOW) : CBOW is a variation of the feedforward NNLM. The weight matrix between the input and the projection layer is shared for all word positions in the same way as in the NNLM. All words get projected into the same position. This model is called "continuous bag of words" because it uses continuous word representations (unlike standard bag of words), that do not rely on word ordering (like standard bag of words).

Skip-gram: Skip-grams are similar to CBOW, but instead of predicting the current word based on the context, it tries to maximize classification of a word based on another word in the same sentence. Each current word as an input to a log-linear classifier with continuous projection layer and predicts words within a certain range before and after the current word. Increasing the range improves quality of the resulting word vectors. Since the more distant words are usually less related to the current word than those close to it, it gives less weight to the distant words by sampling less from those words in the training examples.

As shown in Mikolov et al. (2013b), the objective of the skip-gram model is to maximize the average log probability. Mikolov discusses a variety of approaches to optimize this calculation and to balance accuracy and training examples.

3 Data

As discussed above, I use the "gold-standard" adjectival scales (half and full) from both Wilkinson and Oates (2016) and de Melo and Bansal (2013), as well as generated adjectival scales using the intensity data provided by Taboada et al. (2011). I only include scales that have three or greater adjectives in the scale (since adjective pairs are not too useful to compare for our purposes). I run experiments on both the half scales and the full scales, but have separated the results into (Table 1) for half scales and (Table 2) for full scales.

Oates (Wilkinson and Oates (2016)) This dataset is simply 12 "gold-standard" full adjectival scales ranging in size (from four to seven adjectives) and complexity (defined loosely as a measure of how abstract the adjectives are). An example of a complex scale would be: *same, alike, similar, different*. A simple scale would be: *freezing, cold, warm, hot*. These scales were generated, cleaned, and sourced by crowd-sourcing answers via Mechanical Turk to determine which adjective was "higher" than the other.

Bansal (de Melo and Bansal (2013)) This dataset has an initial 76 "gold-standard" half adjectival scales that have greater than two values. They begin with full scale sets, which are are extracted from clustering WordNet dumbbell structures, extended with synonyms, and then split into two antonymous halves. I recreate the full scales by comparing the poles of different half scales and cross-listing them with WordNet to determine if they are antonyms. If they are, I join the two antonymous half scales. I effectively reverse the process discussed in de Melo and Bansal (2013). This results in 33 "initial" full scales and 1478 extended full scales. I partition this data into four segments: the initial 76 half scale clusters, the extended 261 half scale development set, a recreation of the original full scales, and then a recreation of the extended full scales.

Taboada (Taboada et al. (2011)) Unlike the other two dataset, this dataset has no "gold-standard" adjectival scale. Instead, I try to use this dataset to create my own. The model discussed in Taboada et al. (2011) is focused around analyzing sentiment from text, and grades words based upon intensity (on a scale of 0 to 5) and sentiment (negative numbers indicate a negative opinion, positive numbers indicate a positive opinion). Here Dr. Marianna Apidianaki was indispensable. She

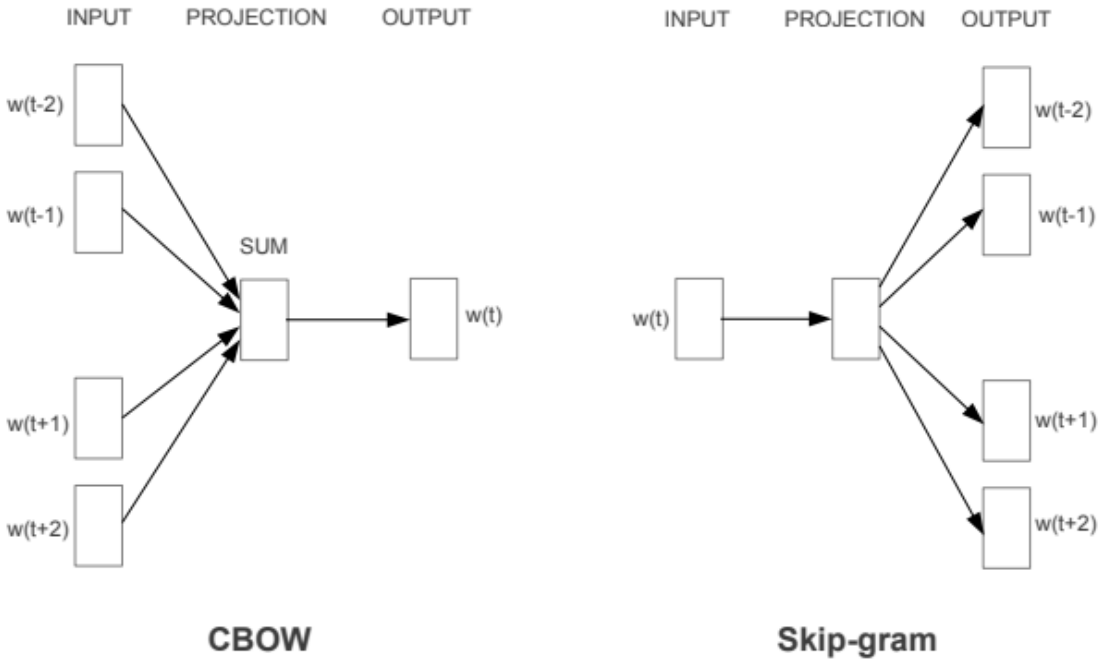

Figure 1: CBOW vs Skip-gram (via Mikolov et al. (2013a))

cross-listed the words in the SO-CAL dictionaries with synonym and antonym sets in WordNet. She then created "intensity pairs," which are words in SO-CAL that are matched with their synonyms or antonyms that are also in SO-CAL. The end result was pairs of related words, with their intensity data. For example: *sinful unholy -2 -3*. Here, *unholy* is considered more negative than *sinful*. I was then able to use those pairings to create both full scales and half scales, based around the intensities of the words (as ranked by SO-CAL). This gives me 673 adjectival half scales and 3163 adjectival full scales of mixed quality.

4 Approach

The approach to this problem is similar to the one observed in Mikolov et al. (2013c) and Mikolov et al. (2013a). That is to say, there exists some relationship between these continuous word representations. Introducing the vector offset method: a method of determining the relationship between continuous word representations based around cosine distance. In this method, there is an assumption of that the relationship presented is a vector offset, so that in the embedding space, all pairs of words sharing a particular relation are related by the same constant offset. In this approach, one can approximate a word $y = x_b - x_a + x_c$. For example, Mikolov et al. (2013c) shows that the word closest (via cosine distance) to $king - man + woman$ is $queen$. Recall that cosine similarity is:

$$w^* = argmax_w \frac{x_w y}{||x_w||||y||}$$

This idea is expanded to adjectives in Kim and de Marneffe (2013). Moreover, the authors present the idea that an intermediate vector can be found between two adjectives, such that the middle vector is the intermediate intensity between the two adjectives. Under the assumption that there exists linear relationship between adjectives, they calculate the "middle" vector: $w_m = w_b + \frac{(w_a - w_b)}{2}$ and then calculate the vector most similar to that midpoint. Figure 2 shows their results (extending the idea to include quartiles) for antonymous adjectives.

My goal is to go just a bit farther, to adapt this approach to the skip-gram (versus an RNNLM) and to experiment with using this method to organize an adjectival scale (versus using the IQAP dataset).

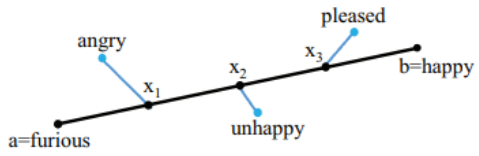

Figure 2: An example of vector with the highest cosine similarity to intermediate points between *furious* and *happy* (via Kim and de Marneffe (2013))

5 Experimentation and Evaluation

5.1 Comparison with Existing Models

In order to evaluate the proposed model, it is worth comparing it with the existing model described in Kim and de Marneffe (2013). Figure 3 and Figure 4 show a side-by-side comparison between the two models.

Figure 3 compares the ranking of the comparative, given a half scale and the mean vector between the adjective and its superlative. I filter out the two input words from the result. It is clear that the RNNLM with 1600 dimensions performs better than the skip-gram NNLM with 300 dimensions. The RNNLM returns the comparative as the first result most of the time, while the NNLM returns the comparative as the third result, or worse. Still, it is clear that there are some similarities between the two, for example both have a dip in performance for finding *worse*.

The NNLM fares better in Figure 4. There, each language model presents the first result for a given quartile in a full adjectival scale. I filter out any words that match the end-points and if a word appears in two different quartiles, I give priority to the quartile it is most similar to. The result is a much fairer assessment, with one glaring exception (*CHEFS_Chefs* in the *hot:cold* scale). In some cases (e.g. the *ugly:gorgeous* scale), the skip-gram out performs even the RNNLM. This seems to suggest that the skip-gram will yield better results for dealing with full adjectival scales, as opposed to half adjectival scales.

5.2 Experiments in Arranging Scales

The purpose of this study is to get a better understanding of the relationship between scalar adjectives in continuous word representations. Many of the studies cited in this paper evaluate the relationship using *question-answer* testing on a dataset, such as IQAP. I arranged a different experiment, with an eye towards the examining the underlying assumption of linearity. I want to know if the relative positioning of adjectives actually tells us what we assume it is telling us.

5.2.1 The Experiment

I construct three variations of the experiment on the different datasets described in the Data section. The crux is this: given the shuffled values of an adjectival scale, can I reconstruct the scale using the relative positions of their vector representations, using the vector offset method discussed above. The three variations are:

E1 - completely shuffled adjectives

E2 - given one random extreme of the scale, the rest of the adjectives are shuffled

E3 - given the two extremes of the scale, the rest of the adjectives are shuffled.

5.2.2 Methodology

I developed three slightly different methods that test the behavior and relationships of the adjective vectors:

M1 - The scale is static points in the continuous space (similar to the static points in the quartiles of Figure 4) and the algorithm is to find the adjectives most similar to those fixed points.

M2 - The scale is more dynamic, where the position vectors of the scale are the words themselves. So the first position vector is the vector of the word most similar to a given extreme, and then the next word in the scale is the word most similar to the previous word vector (as opposed to some fixed linear distance).

M3 - The scale is more loose, focusing only on a given words position relative to any given extreme. This means the constructed scale is centered around the similarity of a word to an extreme, rather than its position between the extremes.

For all methods, if the extremes are given, I calculate the first position via the word most similar to an extreme. If one extreme is given, I calculate the second extreme as the word (out of all the words in the scale) most dissimilar to the given extreme. If no extremes are given, I calculate the two extremes as the words most dissimilar to each other.

5.2.3 Measure of Accuracy

I use Spearman's Rank Correlation ρ to determine the overall effectiveness of the sorting. I com-

Input Words	Comparative	(Kim and de Marneffe, 2013) Ranking	Test Ranking
good:best	better	1	3
bad:worst	worse	4	7
slow:slowest	slower	1	1
fast:fastest	faster	1	3

Figure 3: Words with corresponding vectors closest to the mean of positive:superlative word vectors.

First word (-)	1st quarter	2nd quarter	3rd quarter	Second word(+)
furious	angry - livid	unhappy - unhappy	pleased - glad	**happy**
furious	angry - angry	tense - frantic	quiet - quiet	**calm**
terrible	horrible - horrible	incredible - great	wonderful - fantastic	**terrific**
cold	mild - chilly	warm - frigid	sticky - CHEFS_Chefs	**hot**
ugly	nasty - uglier	wacky - lovely	lovely - beautiful	**gorgeous**

Figure 4: Adjectival scales extracted from the NNLM: each row represent a scale, and for each intermediate point the closest word in term of cosine similarity is given. For each column first word is from Kim and de Marneffe (2013) and second word is from the test set.

pare the predicted scale with the "gold-standard" scale. To borrow an idea from Cocos et al. (2018), for each dataset, I calculate this metric by treating each adjective in a particular scale as a single data point, and calculating an overall ρ for all adjectives from all scales. For all of the experiments, I do not include the given extreme value (if one is given) when calculation the Spearman's Rank Correlation, in order to give the most accurate assessment of the methodology. I also ensure that the different algorithms have to sort at least two values on the scale (to prevent artificial inflation of the score by sorting one value, when the two extremes are given).

5.2.4 Results

Figures 5, 6, 7 shows the results of methods E1, E2, E3 for M1, M2, and M3 respectively. I removed any scales where the word is not in the NNLM and/or the Spearman correlation computed is NaN.

6 Discussion and Future Work

I wish I would have been able to put more of myself into this work. In better circumstances, I would have like to delve a little bit deeper on the NNLM side of things and poke around. There is definitely some relationship between continuous word representations, but I think it is much more complicated than Kim and de Marneffe (2013) lets on. While you can apply a simple linear projection of the vectors to get you in the somewhat right direction, it can also be deceptive. Depending on the

M1 Dataset	E1	E2	E3
Oates_half	.548	.644	.964
Oates_full	.469	.469	.401
SOCAL_half	.683	.565	.546
SOCAL_full	.360	.142	.101
deMelo_half_init	.689	.631	.476
deMelo_half_EXT	.516	.425	.354
deMelo_FULL_init	.381	.268	.314
deMelo_full_EXT	.324	.186	.191

Figure 5: M1 - Spearman Correlation for M1 (computing similarity to static scale)

.

M2 Dataset	E1	E2	E3
Oates_half	.559	.647	.750
Oates_full	.568	.408	.824
SOCAL_half	.691	.584	.616
SOCAL_full	.361	.161	.231
deMelo_half_init	.701	.665	.634
deMelo_half_EXT	.524	.439	.389
deMelo_FULL_init	.256	.232	.278
deMelo_full_EXT	.351	.225	.275

Figure 6: M2 - Spearman Correlation for M2 (computing similarity to dynamic scale)

.

training set, the similarity of adjective vectors can reflect anything from intensity, to type of adjective (comparative, superlative, etc.), to even antonyms (in fact, often times antonyms are most similar to each other). Unfortunately, this study was full of self-inflicted set-backs. I want to thank you for

M3 Dataset	E1	E2	E3
Oates_half	.559	.647	.964
Oates_full	.597	.474	.881
SOCAL_half	.689	.585	.609
SOCAL_full	.377	.193	.265
deMelo_half_init	.716	.688	.719
deMelo_half_EXT	.522	.417	.423
deMelo_FULL_init	.352	.302	.372
deMelo_full_EXT	.351	.233	.285

Figure 7: M3 - Spearman Correlation for M3 (computing similarity to extremes)

.

your patience. It has been a long journey.

References

Anne Cocos, Veronica Wharton, Ellie Pavlick, Marianna Apidianaki, and Chris Callison-Burch. 2018. Learning scalar adjective intensity from paraphrases. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1752–1762.

Joo-Kyung Kim and Marie-Catherine de Marneffe. 2013. Deriving adjectival scales from continuous space word representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1625–1630.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *ICLR Workshop*.

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech*, pages 1045–1048.

Tomas Mikolov, Daniel Povey, Lukas Burget, and Jan Cernocky. 2011. Strategies for training large scale neural network language models. *In Proceedings of ASRU*, pages 196–201.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*, pages 746–751.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stedel. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics 2011 Vol. 37*, pages 267–307.

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)*.