Defining Partial Order over Adjectives Clusters

Xiao Ling
lingxiao@seas.upenn.edu

Abstract

Adjectives such as good, great, and excellent are similar in meaning but differ in intensity. Intensity ordering is useful in several NLP tasks, and in general defining any algebra over some subset of lexicon is an important first step in properly characterizing the semantics of a language. However this data is missing in most lexical resources such as dictionaries and WordNet. In this paper we present an unsupervised apporach that first pairwise rank adjectives by approximating their distribution around select liguistic-patterns, then resolve inconsistencies using an integer linear programming formulation. We test our approach on the English adjective clusters distributed by de Melo and Bansal (2013), achieving 75.0% pairwise accuracy without relying on annotator information as Bansal did. Most notably, we broke through the 0.60 Kendall's tau barrier eluding previous research, thereby achieving near human-level performance under Kendall's tau and pairwise accuracy.

1 Introduction

Linguistic scale is a set of words of the same grammatical category that can be ordered by their expressive strength or degree of informativeness (Sheinman and Tokunaga, 2009). Ranking adjectives over such a scale is a particularly important task in sentiment analysis, recognizing textual entailment, question answering, summarization, and automatic text understanding and generation. For instance, understanding the word "great" is a stronger indicator of quality than the word "good" could help refine the decision boundary between four star reviews versus five star one. However, current lexical resources such as WordNet do not provided such crucial information about the intensity order of adjectives.

Past work approached this problem in two ways: distributional and linguistic-pattern based. Kim and de Marneffe (2013) showed that word vectors learned by a recurrent neural network language model can determine scalar relationships among adjectives. Specifically, given a line connecting a pair of antonyms, they posited that intermediate adjective word vectors extracted along this line should correspond to some intensity scale determined by the antonyms. The quality of the extracted relationship is evaluated using indirect yes/no question answer pairs, and they achieved 72.8% pairwise accuracy over 125 pairs.

While distributional methods infer pairwise relationship between adjectives based on how they occur in the corpus separately, linguistic-pattern based approaches decides this relationship using their joint co-occurence around pre-determined patterns (Sheinman and Tokunaga, 2009; Schulam and Fellbaum, 2010; Sheinman et al., 2012) . For example, the phrase "good but not great" suggests good is less intense than great. These patterns are hand-curated for their precision and unsurprisingly enjoy high accuracy. However, they suffer from low recall because the amount of data needed to relate a pair of adjectives is exponential in length of the pattern, while such patterns are no less than four to five words long.

de Melo and Bansal (2013) addressed this data sparsity problem by exploiting the transitive property of partial orderings to determine unobserved pairwise relationships. They observed that in order to deduce an ordering over good, great, and excellent, it suffices to observe good is less than great, and great is less than excellent. Then by transitive property of the ordering we conclude good is also less than excellent. This fixed relationship among adjectives is enforced by a mixed integer linear program (MILP). Banal and de Melo tested their approach on 91 adjective clusters, where the average number of adjectives in each cluster is just over three, and each cluster is ranked by a set of annotators. They reported 69.6% pair-wise accuracy and 0.57 average Kendall's tau.

Bansal's method suffer from one major drawback, observe in the example above if we only observe an ordering between good and great, and good and execellent, then no conclusion can be made about the ordering between great and excellent. In general, in order to place an ordering over n items, we need $n-1$ "critical" pairwise comparisons. This is a very restrictive assumption in practice, in fact most adjectives simply do not co-occur around the given set of patterns at all, thus no meaningful ordering may be placed. We combat the data sparsity problem in three ways. First, we extract additional linguistic pattern. Next we assume the phrases are generated in a Markov manner to approximate the probability of unseen phrases. Finally, we explore novel sources of data extracted from PPDB corpus. Once the data is prepared, we use a variety of inference methods to rank the adjectives, revealing the promises and limitations of each dataset and method.

2 Data Preparation

This section contains a detailed description of how the data is procured and preprocessed, as well as how the training and test sets are created. For ease of reproduction, all data used for this paper is distributed in the project source file (https://github.com/lingxiao/good-great).

2.1 Extracting Intensity Patterns

Both de Melo and Bansal (2013) and Sheinman et al. (2012) showed that linguistic patterns connecting two adjectives reveal semantic intensities of these adjectives. Sheinman extracted the patterns by first compiling pairs of seed words where the relative intensity between each pair is known. Then they collected patterns of the form "a * b" for each pair from an online search engine, where * is a wildcard denoting one or more words, and word "a" is fixed to be weaker than word "b". Sheinman then took the intersection of all wildcard phrases appearing between all pairs of words, thereby revealing a set of "weak-strong" patterns P_{ws} where words appearing in front of the pattern is always weaker than the word appearing behind. Table 2 shows the weak-strong patterns extracted by Sheinman. Bansal used a similar approach but used the Google N-gram corpus (Brants and Franz, 2006) as the source of patterns. Additionally, they also considered "strong-weak" patterns P_{sw} where words appearing in front of the pattern are stronger than those appearing behind. See table 1 for the set of strong-weak and weak-strong patterns mined by Bansal. Finally, during the course of the project, we found additional strong-weak patterns in the N-gram corpus that increased the accuracy of our results, they are found in table 2.

Strong-Weak Patterns	Weak-Strong Patterns
not * (,) just *	* (,) but not *
not * (,) but just *	* (,) if not *
not * (,) still *	* (,) although not *
not * (,) but still *	* (,) though not *
not * (,) although still *	* (,) (and,or) even *
not * (,) though still *	* (,) (and,or) almost *
* (,) or very *	not only * but *
not * (,) just *	not just * but *

Table 1: Bansal and de Melo's linguistic patterns. Note the syntax (and,or) means either one of "and" or "or" are allowed to appear, or not appear at all. Similarly, (,) denotes a comman is allowed to appear. Additionally, articles such as "a", "an", and "the" are may also appear before the wildcards. Wildcards matches any string.

Strong-Weak Patterns	Weak-Strong Patterns
* (,) unbelievably *	very * (and,or) totally *
* not even *	* (,) yet still *
	* (,) (and,or) fully *
	* (,) (and,or) outright *

Table 2: The weak-strong patterns were found by Sheinman. We mined for the strong-weak patterns from google N-gram corpus.

2.2 Collecting Pattern Statistics from N-grams

We used the Google N-gram Web 1T 5-gram Version 1 publicly distributed by the Linguistic Data Consortium to replicate Bansal's results. Because we aggressively downsized the N-gram corpus, a detail account of our process is given here. The entire N-gram corpus was first normalized by case folding and white-space stripping. Then for each linguistic pattern in tables 1 and 2, we grepped the corpus for key words appearing in each pattern. Both the grep commands and their corresponding grepped ngrams are located in the raw-data directory of project folder. The grepped ngram corpus is several times smaller than the original corpus, thus dramatically increasing the number of experiments we can perform.

Next, we crawled the grepped corpus for the patterns found in tables 1 and 2. Specifically, for each pattern of form $*P*$ and pairs of words a_1 and a_2, we collect statistics for a_1Pa_2 and a_2Pa_1. In a departure from Bansal's method, we also collected statistics for $*Pa_1$, $*Pa_2$, a_1P*, and a_2P*, where $*$ is allowed to be any string. Finally, we also count the occurences of each pattern $*P*$.

2.3 Extracting PPDB Adverb Patterns

While de Melo and Bansal (2013) and Sheinman et al. (2012) only considered patterns that relate pairs of adjectives to each other, we also considered adverbs and adverb phrases that occur infront of adjectives, thereby modifying their intensity. We hypothesized that the adverbs can be roughly separated into three classes: intensifying, deintensifying, and netural. For example, we suspect the adverb "extremely" might intensify adjectives such as "good" in the phrase "extremely good", while "slightly" would deintensify adjectives it modifies. In general however, neither the class in which adverbs belong to nor the degree in which they modify adjectives are clear, thus both need to be learned from corpora.

The paraphrase database (PPDB) (Pavlick et al., 2015) makes this learning problem possible. This database maps english utterances to other english utterances of similar meaning, so that if (x_1, x_2) appears in the database, then we conclude x_1 and x_2 are paraphrases. Section 3.3 outlines a method that uses (adverb-adjective, adjective) and (adjective, adverb-adjective) pairs to simultaneously assign scalar values to both the adjectives and adverbs for the task of adjective ranking. We test our assignment on pairs of adjectives ranked by Amazon mechanical turks that also appear in the PPDB corpus. In order to reduce noise in the labels, we removed pairs of adjectives where there is no simple majority consensus among the turks.

3 Problem Formulation

This section presents three formulations of the ranking problem. We show derivations when possible, otherwise qualitative argument is given.

3.1 Global Ranking with Two Sided Patterns

Bansal and de Melo de Melo and Bansal (2013) use pairwise co-occurence of adjective pairs around the paraphrases found in table 1 to infer the relative strength between the adjectives. Because this data is missing for most pairs, they confronted this problem by computing pairwise rankings when possible, and using the transitive property of partial rankings to infer the missing relationships. Pairwise ranking is computed as:

$$score(a_1, a_2) = \frac{(W_1 - S_1) - (W_2 - S_2)}{cnt(a_1) \cdot cnt(a_2)},$$

where:

$$W_1 = \frac{1}{P_1} \sum_{p \in P_{ws}} cnt(p(a_1, a_2))$$

$$W_2 = \frac{1}{P_1} \sum_{p \in P_{ws}} cnt(p(a_2, a_1))$$

$$S_1 = \frac{1}{P_2} \sum_{p \in P_{sw}} cnt(p(a_1, a_2))$$

$$S_2 = \frac{1}{P_2} \sum_{p \in P_{sw}} cnt(p(a_2, a_1)),$$

with:

$$P_1 = \sum_{p \in P_{ws}} cnt(p(*, *))$$

$$P_2 = \sum_{p \in P_{sw}} cnt(p(*, *)).$$

Observe W_1 measures the likelihood of encountering the phrase a_1pa_2 conditioned on the fact that the corpus is composed entirely of phrases of form $*p*$; a similar interpretation holds for W_2, S_1, and S_2. Furthermore, $(W_1 - S_1) - (W_2 - S_2)$ is positive when a_1 occurs more often on the weaker side of the intensity scale relative to a_2, hence $score(a_1, a_2)$ is an $cardinal$ measure how much weaker a_1 is relative to a_2. The denominator $cnt(a_1) \cdot cnt(a_2)$ penalizes high absolute value of the numerator due to higher frequency of certain words, thus normalizing the score over all pair of adjectives, and therefore global comparison is well defined over some cardinal scale. Finally, observe that $score(a_1, a_2) = -score(a_1, a_2)$.

Given pairwise scores over a cluster of adjectives where a global ranking is known to exist, Bansal then aim to recover the ranking using mixed integer linear programming. Assuming we are given N input words $A = \{a_1, ..., a_N\}$, the MILP formulation places them on a scale $[0, 1]$ by assigning each a_i a value $x_i \in [0, 1]$. The objective function is formulated so that if $score(a_i, a_j)$ is greater than zero, then we know a_i is weaker than a_j and the optimal solution should have $x_i < x_j$. The entire formulation is reproduced below:

Maximize

$$\sum_{i,j} (w_{ij} - s_{ij}) \cdot score(a_i, a_j)$$

s.t

$$\begin{aligned}
d_{ij} &= x_j - x_i && \forall i, j \in \{1, ..., N\} \\
d_{ij} - w_{ij}C &\leq 0 && \forall i, j \in \{1, ..., N\} \\
d_{ij} + (1 - w_{ij})C &> 0 && \forall i, j \in \{1, ..., N\} \\
d_{ij} + s_{ij}C &\geq 0 && \forall i, j \in \{1, ..., N\} \\
d_{ij} - (1 - s_{ij})C &< 0 && \forall i, j \in \{1, ..., N\} \\
x_i &\in [0, 1] && \forall i, j \in \{1, ..., N\} \\
w_{ij} &\in \{0, 1\} && \forall i, j \in \{1, ..., N\} \\
s_{ij} &\in \{0, 1\} && \forall i, j \in \{1, ..., N\}.
\end{aligned}$$

Note d_{ij} captures the difference between x_i and x_j, C is a very large constant greater than $\sum_{i,j} |score(a_i, a_j)|$. If the variable $w_{ij} = 1$, then we conclude $a_i < a_j$, and vice versa for s_{ij}. The objective function encourages $w_{ij} = 1$ for $score(a_i, a_j) > 0$ and $w_{ij} = 0$ otherwise. Furthermore, note either s_{ij} or w_{ij} can be one, thus the optimal solution does not have ties. Bansal then extended the objective to incorporate synonymy information over the N adjectives, defined by $E \subseteq \{1, ..., N\} \times \{1, ..., N\}$. The objective is now to maximize:

$$\sum_{(i,j) \notin E} (w_{ij} - s_{ij}) \cdot score(a_i, a_j) - \sum_{(i,j) \in E} (w_{ij} + s_{ij}) \cdot C,$$

while the constraints remain unchanged. The additional set of terms encourages both s_{ij} and w_{ij} to be zero if both a_i and a_j are in E, thus the optimal solution may contain synonyms. We discuss the benefits and draw back of this approach in section four, but for now it suffices to say that in practice we observe $score(a_i, a_j) = 0$ for most pairs of adjectives within a cluster, this data sparsity motivates our next formulation.

3.2 Pairwise Ranking with One Sided Patterns

Since $score(a_i, a_j)$ is zero for most pairs of adjectives due to lack of data, we are motivated to find alternate ways of approximating this value using single sided patterns of form: $\{a_ip*, a_jp*, *pa_i, *pa_j\}$. Loosely speaking, even if we do not observe any $a_i\ p\ a_j$ for some weak-strong pattern p, we can still approximate the liklihood of observing this string using the frequency in which a_i appears in front of the the pattern p, and the frequecy a_j appears behind the pattern p. Once we approximate the liklihood for $a_j\, pa_i$ over weak-strong patterns p, and similarly for all strong-weak patterns, we can infer whether adjective a_i is weaker than a_j by determining whether a_i is more likely to appear on the weaker side of each phrase. This intuition is naturally expressed in the heuristic:

$$score(u) = \frac{cnt(u\, p_{sw}\, *) + cnt(*\, p_{ws}\, u)}{cnt(u\, p_{ws}\, *) + cnt(*\, p_{sw}\, u)},$$

where:

$$cnt(u\, p_{sw}\, *) = \sum_{v \in \mathbf{V}} \sum_{p \in P_{sw}} cnt(u\, p\, v).$$

This score captures the proportion of times u dominates all other words through the patterns given, relative to the proportion of times u is dominated by all other words through the pattern. The results of this ranking is displayed in table 6 in the line "Markov heuristic."

Now we make the intuition precise. Suppose we have a simple language $\mathbf{L}$ made up only of phrases of the form "word pattern word" for every word in the unigram set $\mathbf{V}$ and every pattern in table 1, that is we have:

$$\mathbf{L} = \{u\,p\,v \quad | \quad u, v \in \mathbf{V}, p \in \mathbf{P_{sw}} \cup \mathbf{P_{ws}}\}.$$

If we can confidently approximate liklihood of each phrase from $\mathbf{L}$ based on N-gram corpus evidence alone then we are done. But because data is sparse, we must fill in the missing counts by assuming the phrases in $\mathbf{L}$ is generated by this markov process involving two random variables, V whose value range over vocabulary $\mathbf{V}$, and P ranging over the patterns in table 1. The speaker selects a word u according to some distribution $\mathcal{D}_V$ over $\mathbf{V}$, then conditioned on the fact that u is drawn, a phrase p is drawn according to the conditional distribution $\mathcal{D}_{P|V}$. Finally, conditioned on the fact that p is drawn, a word v is sampled from $\mathcal{D}_{V|P}$. The attentive reader will observe that this crude model does not respect word order. In the phrase "not great (,) just good", our model would generate the phrase "good not just great". Surpringly this model works well enough to outperform Bansal' method. Now the probabilty of a phrase is:

$$\mathcal{D}_L = \frac{\mathcal{D}_V\, \mathcal{D}_{P|V}\, \mathcal{D}_{V|P}}{Z},$$

where Z is an appropriate normalization constant. But since we are only interested comparing the relative liklihood of phrases, Z doese not need to be computed. So we have:

$$\mathcal{D}_L = Pr[u\,p\,v] \propto Pr[u]Pr[p|u]Pr[v|p], \quad (1)$$

where:

$$Pr[V = u] = \frac{cnt(u)}{cnt(*)}$$

$$Pr[P = p|V = u] = \frac{cnt(u\,p\,*)}{cnt(u\,*)}$$

$$Pr[V = v|P = p] = \frac{cnt(*\,p\,v)}{cnt(*\,p\,*)}$$

where $cnt(*) = \sum_{x \in \mathbf{V}} cnt(x)$. The first distribution is approximated by the one-gram corpus, the second and third distribution by four and five grams. In the interest of not computing normaliztion constant whenever possible, we put the following crude bound on $cnt(u\,*)$:

$$cnt(u\,*) = \sum_x count(u\,x) \leq cnt(u),$$

where x is ranges over all suffixes of length three or four. So (1) becomes:

$$Pr[u\,p\,v] \propto \tfrac{cnt(u\,p\,*) \cdot cnt(*\,p\,v)}{cnt(*) \cdot cnt(*\,p\,*)}. \quad (2)$$

Now define the probability that u is stronger than v under $\mathcal{D}_L$ as:

$$\begin{aligned} Pr[u \; > \; v] &= Pr[u\,P_{sw}\,v \quad or \quad v\,P_{ws}\,u] \\ &= Pr[u\,P_{sw}\,v] + Pr[v\,P_{ws}\,u] \\ &= \sum_{p \in P_{sw}} Pr[u\,p\,v] + \sum_{p \in P_{ws}} Pr[v\,p\,u], \end{aligned}$$

and similarly for $v \; > \; u$. We decide u is stronger than v if:

$$\begin{aligned} & Pr[u \; > \; v] \geq Pr[v \; > \; u] \\ & \Longrightarrow \\ & \frac{cnt(uP_{sw}*) \cdot cnt(*P_{sw}v)}{cnt(*) \cdot cnt(*P_{sw}*)} + \frac{cnt(vP_{ws}*) \cdot cnt(*P_{ws}u)}{cnt(*) \cdot cnt(*P_{ws}*)} \\ & \geq \\ & \frac{cnt(vP_{sw}*) \cdot cnt(*P_{sw}u)}{cnt(*) \cdot cnt(*P_{sw}*)} + \frac{cnt(uP_{ws}*) \cdot cnt(*P_{ws}v)}{cnt(*) \cdot cnt(*P_{ws}*)} \\ & \Longrightarrow \\ & \frac{cnt(uP_{sw}*) \cdot cnt(*P_{sw}v)}{cnt(*P_{sw}*)} + \frac{cnt(vP_{ws}*) \cdot cnt(*P_{ws}u)}{cnt(*P_{ws}*)} \\ & \geq \\ & \frac{cnt(vP_{sw}*) \cdot cnt(*P_{sw}u)}{cnt(*P_{sw}*)} + \frac{cnt(uP_{ws}*) \cdot cnt(*P_{ws}v)}{cnt(*P_{ws}*)}. \end{aligned} \quad (3)$$

Note the normalization constant $cnt(*)$ drops out, and there is a qualitative symmetry in (3) that echos intuition. Since (3) does not output cycles, ranking is done by topological sort; results are reported in table 6 under "markov pairwise approximate." Next, we combine the approximate value from (3) with those directly observed in the corpus. If for some adjective pair (u, v) we observe any one of the following values: $u\, p_{sw}\, v$, $u\, p_{ws}\, v$, $v\, p_{sw}\, u$, or $v\, p_{ws}\, u$, then we can define the probability that $u > v$ under $\mathcal{D}_L$ as:

$$Pr[u > v] = \frac{cnt(u\; P_{sw}\; v) + cnt(v\; P_{ws}\; u)}{Z} \qquad (4)$$

where

$$\begin{aligned} Z = cnt(u\; P_{sw}\; v) + cnt(v\; P_{ws}\; u) \\ + cnt(v\; P_{sw}\; u) + cnt(u\; P_{ws}\; v), \end{aligned}$$

and

$$cnt(u\; P_{sw}\; v) = \sum_{p \in P_{sw}} cnt(u\, p\, v).$$

Now to rank u against v, we compute (4) if possible, otherwise we approximate the probability that $u > v$ using (3). Since the ordering over each pair of adjectives is decided separately using (3) or (4), cycles do exist and transitivity must be enforced. First, we consider a simple integer linear programming formulation, given N adjectives in a cluster where a ranking is known to exist, and define:

$$P_{uv} = \frac{Pr[u > v]}{Pr[v > u]},$$

so that if $u \geq v$ under $\mathcal{D}_L$ then $P_{uv} \geq 1$:

$$\begin{aligned} &\textbf{Maximize} \\ &\sum_{u,v \in \{1,..,N\}} P_{uv} \cdot s_{uv} + P_{vu} \cdot (1 - s_{vu}) \\ &\textbf{s.t} \\ &(1 - s_{uv}) + (1 - s_{vw}) \geq (1 - s_{uw}), \\ &\forall u, v, w \in \{1, ..., N\}. \end{aligned}$$

unknown adverb	**unknown adjective**
great,greater,very	little,meager,very
great,greatest,really	warm,hot,really

Table 3: An example from the PPDB corpus. In column 1 we initialize the base, comparative and superlative triples to some value in [-1,1] and solve for values of the adverbs. In column 2 we are given adjectives of unknown value, and use the now known adverbs to determine their value.

Thus the objective encourages $s_{uv} = 1$ if $u > v$, and $s_{uv} = 0$ otherwise. Overall the objective gives precedent to those pairs where u dominates v the most, while the constraints enforce transitive properties of the ordering. This mixed approximation of $Pr[u > v]$ and ILP gives the best results on Bansal's data, see tables 4 and 5 under "Markov pairwise mixed ILP." Lastly, in the interest of exploring the trade off between precision versus sparsity of data, we use our approximation of frequency of $u\,p\,v$ in Bansal's formulation. Define:

$$\begin{aligned} score(u, v) = Pr[v\; P_{sw}\; u \quad or \quad u\; P_{ws}\; v] \\ - Pr[u\; P_{sw}\; v \quad or \quad v\; P_{ws}\; u], \end{aligned}$$

so that $score(u, v) > 1$ if $u < v$ under $\mathcal{D}_L$, thereby conforming to Bansal's formulation of the score function in terms of sign. See "Markov pairwise approximate MILP."

3.3 Pairwise Ranking using learned category of adverbs.

Algorithm 1: Simultaneous assignment of adjective and adverb intensities.

```
function Assign ($PPDB$);
Input  : PPDB paraphrases
Output: $\gcd(a, b)$
if $b = 0$ then
    return $a$;
else
    return Euclid($b, a \bmod b$);
end
```

All solutions above share two common drawbacks: the use of phrases of at least length three, which introduces data sparsity problem, and hard assignment of phrases to their respective categories, which may not always be justified. For instance we assume all words appearing in the second wildcard position of the phrase "* (,) or very *" are weak words, because a priori we assume "or very" must be followed by a weaker word. In this section we sidestep the data sparsity problem by only considering bigrams of form "adverb adjective" for some set of adverbs, and engage the assignment of categories problem by learning a scalar value for each adverb from corpus, denoting the degree in which they modify the intensity of the following adjective. Our method is shown in algorithm 1.

4 Results

In this section, we discuss the three measures used to evaluate our results, and then compare our approaches against such metrics.

4.1 Metrics

First we consider pairwise accuracy as defined by Bansal de Melo and Bansal (2013). If every adjective in each cluster is assigned a numerical ranking $r(a_i)$, then the label of each pair is defined as as:

$$L(a_i, a_j) = \begin{cases} > & if \quad r(a_i) > r(a_j) \\ < & if \quad r(a_i) < r(a_j) \\ = & if \quad r(a_i) = r(a_j). \end{cases}$$

Given gold-standard labels L_G and predicted labels L_P, the pairwise accuracy of each cluster of adjectives is the fraction of pairs that are correctly classified:

$$PW = \frac{\sum_{i<j} \mathbb{1}(L_G(a_i, a_j) = L_P(a_i, a_j))}{\sum_{i<j} \mathbb{1}}$$

Next we assess the rank correlation between the gold-standard and the predicted set. Kendall's tau measures the total number of pairwise inversions Kruskal (1958):

$$\tau = \frac{P - Q}{\sqrt{(P + Q + X)(P + Q + Y)}}. \quad (5)$$

P measures the number of concordant pairs and Q is the number of discordant pairs, X is the number of pairs tied in the gold ranking, and Y is number of ties in the predicted ranking. The pair (ai, aj) are:

- concordant if $r_G(a_i) < r_G(a_j)$ and $r_L(a_i) < r_L(a_j)$ or $r_G(a_i) > r_G(a_j)$ and $r_L(a_i) > r_L(a_j)$
- discordant if $r_G(a_i) < r_G(a_j)$ and $r_L(a_i) > r_L(a_j)$ or $r_G(a_i) > r_G(a_j)$ and $r_L(a_i) < r_L(a_j)$
- tied if $r_G(a_i) = r_G(a_j)$ and $r_L(a_i) = r_L(a_j)$ or $r_G(a_i) = r_G(a_j)$ and $r_L(a_i) = r_L(a_j)$

Since many of our ranking methods do not allow ties, we also consider a variant where the ties are not counted:

$$\tau' = \frac{P - Q}{n \cdot (n-1)/2}, \quad (6)$$

here n is the number of adjectives in a cluster, and $\frac{n \cdot (n-1)}{2}$ is the total number of unique pairs. In this case the predicted label is discordant w.r.t. gold if the label is flipped, or if the gold-standard pair is a tie. The overall efficacy of each ranking method is captured by finding the average kendall's tau score. Additionally, Bansal observed that sometimes the ordering of adjectives was clear but the annotators would disagreed about which end of the scale was the stronger one, thus absolute kendall's tau is also reported.

During the course of this project we observed that it is possible to outperform certain gold standards under (5). This behavior is highly unexpected and it behooves the reader to consider this concrete example. Suppose our gold standard is: $G = [[a, b], [c]]$, read as: a is tied with b, and they both dominate c. Then relative to itself, G has three unique pairs:

$$pairs = [(a, b), (a, c), (b, c)],$$

two concordant pairs: $P = [(a, c), (b, c)]$, no discordant pairs, and one tied pair so that both X and Y in (5) are one. Thus Kendall's tau of G with respect to itself is:

$$\begin{aligned} \tau &= \frac{2 - 0}{\sqrt{(2 + 0 + 1)(2 + 0 + 1)}} \\ &= \frac{2}{\sqrt{9}} = \frac{2}{3}. \end{aligned}$$

Gold	**MILP**	**Markov ILP**
(cool, chilly) $<$ unfriendly $<$ hostile	unfriendly $<$ cool $<$ hostile $<$ chilly	cool $<$ chilly $<$ unfriendly $<$ hostile
strong $<$ intense $<$ terrible $<$ overwhelming $<$ violent	strong $<$ intense $<$ terrible $<$ overwhelming $<$ violent	intense $<$ strong $<$ terrible $<$ violent $<$ overwhelming
high $<$ higher $<$ soaring	soaring $<$ high $<$ higher	soaring $<$ high $<$ higher

Table 4: The top row displays an example where Bansal's MILP fails to output the correct order (tau = -0.18) but Markov Mixed ILP output the correct order modulo ties (tau = 0.91). The middle row is an example where Bansal' MILP correctly predicted the ranking despite sparse data, only six out of twenty pairs had any N-gram hits. Using markov assumption the missing data was filled in, but at the cost accuracy. The bottom row shows an instance where both methods fail because there is overwhelming copora evidence that higher is more intense than soaring, revealing the limitations of the pattern-based approach.

Observe in the case of ties, the maximum Kendall's tau is less than 1. Next consider the ranking $A = [[a], [b], [c]]$, read as a dominates b and c, while b dominates c. Once again we have two concordant pairs $[(a, c), (b, c)]$ but no discordant pairs by definition, $X = 1$ and now $Y = 0$ because A does not have ties. Thus A with respect to G is:

$$\begin{aligned}\tau &= \frac{2-0}{\sqrt{(2+0+1)(2+0)}} \\ &= \frac{2}{\sqrt{6}} > \frac{2}{3}.\end{aligned}$$

Therefore an algorithms that ranks the adjectives in correct order without ties can actually outperform the gold standard against itself if the gold ranking does have ties. In the interest of fair comparision, we also report how well gold performs against itself in table 5.

4.2 Analysis

table 5 reports all results from the different approaches, while table 6 reports results from all approaches incoporating the new patterns from table 2. Now we discuss the advantanges and drawbacks of each method, specifically the settings in which each method succeeded over another.

First, note that our reproduction of Bansal's MILP is accurate with acceptable errors. However, we were not able to reproduce MILP with synonymy accurately because Bansal relied on synonyms marked by annotators, these annotations were not released in the public codebase accompanying the paper. Instead we attempted to replicate the experiment using synonyms used by wordnet, and observed a marked decrease in accuracy across all measures.

Looking at table 6 line "markov pairwise mixed ILP", we see that the integer linear programming formulation with missing data approximated by (3) enjoy the highest accuracy across all measures. In particular the average tau breaks through the 0.6 barrier and is only six percent away from inter-annotator agreement tau of 0.67. Pairwise accuracy is also very close to inter-annotator agreement. Thereby validating the hypothesis that the markov condition is sufficiently reasonable assumption to approximate frequency of unseen phrases. Furthermore, it is interesting that "markov pairwise approximate" performs almost as well as Banal's MILP method, even though it relies on the approximation given by (3) alone. This indicates that the linguistic patterns specified is an accurate enough of a predictor of relative adjective strength, so that even crude approximations are powerful predictors. In terms of data complexity, we stress that both (3) and (4) can be calculated from the same data collected by Bansal to compute the score in section 3.1, suggesting that the aforementioned data holds more information about the ordering of adjectives than is used by Bansal' MILP method. Most notably, we outperform Bansal' method without relying on annotator labels of synonyms, which a very restrictive condition indeed.

Next we compare the results of table 5 to that of 5. Recall in table 6 we incorporated six additional phrases from table 2, we see all methods benefited from the additional patterns, which increased counts of phrases without sacrificing accuracy. However "markov pairwise mixed ILP" benefited the most across all measures, suggesting that so long as transitivity is enforced, more data leads to better results.

Now we wish to highlight some specific instances where "markov pairwised mixed ILP" did well relative to Bansal' MILP method, and where it faltered. In the interest of fairness, we compare the output of our methods from table 6 only. Bansal's MILP method output 11 rankings where Kendall's tau score was less than zero, while markov pairwise mixed ILP only output six. Notably, five out of six bad ILP clusters also appear in Bansal' method, one example of this is given in the bottom row of table 4. In all these cases there is strong corpora evidence leading to the wrong conclusion, but in 94% of the cases linguistic-pattern points in the right direction. Next we say a ranking is "average" if the tau score is between 0 and 0.6. 37 out of 91 ranking in Bansal's method are just average, while 31 of markov ILP are average. Markov ILP moved 10 clusters from the average range to the "good" range, where tau is greater than 0.6. A tau score is "great" if it is above 0.8, 43 of Markov ILP's rankings are great, while only 35 of MILP's are in this range. Finally, 20 out of 91 rankings output by MILP achieved a perfect Kendall's tau score, while ILP achieved perfect in 26 clusters. This suggests that we can expect the markov approximation of pattern co-occurences to improve each category so long as transitivity is enforced. Finally we also found that Bansal's score display a higher variance of 0.18, while Markov ILP has variance 0.16, suggesting that our method is more concentrated around the true rankings. Refer to table 4 for specific instances where each method shines.

5 Conclusion

In this paper, we dramatically improve de Melo and Banal's work at ranking adjective in terms of their intensity on a continuous scale. We confront sparsity of data in three ways: leveraging webscale corpus, incorporating additional linguistic patterns, and most importantly filling in missing data by approximating the distribution of phrases by assuming the phrases are generated in a Markov manner. Since the approximation is not gauranteed to be consistent, we enforce transitivity using a simple integer linear program.

Not only is our integer linear program formulation simpler than that of Bansal's, we also outperform the state of the art across all measures in a nontrivial manner, and approach human level agreement. We do so without relying on additional annotations during ranking, which is a very restrictive assumption. Finally, the sample complexity of our approach is comparable to that of Bansal's.

References

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram Version 1.

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.

Joo-Kyung Kim and Marie-Catherine de Marneffe. 2013. Deriving adjectival scales from continuous space word representations. In *EMNLP*, pages 1625–1630.

William H. Kruskal. 1958. Ordinal measures of association. In *Journal of the American Statistical Association*.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China, July. Association for Computational Linguistics.

Peter F Schulam and Christiane Fellbaum. 2010. Automatically determining the semantic gradation of german adjectives. In *KONVENS*, pages 163–167. Citeseer.

Vera Sheinman and Takenobu Tokunaga. 2009. Adjscales: Visualizing differences between adjectives for language learners. *IEICE TRANSACTIONS on Information and Systems*, 92(8):1542–1550.

Vera Sheinman, Takenobu Tokunaga, Isaac Julien, Peter Schulam, and Christiane Fellbaum. 2012. Refining wordnet adjective dumbbells using intensity relations. In *Sixth International Global Wordnet Conference*, pages 330–337.

Method	**Pariwise Accuracy**	**Avg. τ**	**Avg. $\|\tau\|$**	**Avg. τ'**	**Avg. $\|\tau'\|$**
Inter-Annotator Agreement	78.0%	0.67	0.76	N/A	N/A
Gold Standard	100.0%	0.90	0.90	0.90	0.90
MILP reported	69.6%	0.57	0.65	N/A	N/A
MILP with synonymy reported	78.2%	0.57	0.66	N/A	N/A
MILP reproduced	68.0%	0.55	0.64	0.41	0.54
MILP with synonymy reproduced	65.0%	0.43	0.58	0.31	0.50
Markov heuristic	65.0%	0.43	0.61	0.31	0.52
Markov pairwise approximate	70.0%	0.53	0.63	0.41	0.54
Markov pairwise mixed ILP	72.0%	0.57	0.64	0.44	0.54
Markov MILP	70.0%	0.53	0.65	0.41	0.56

Table 5: Main results using Bansal's patterns. Note τ refers to kendall's τ with ties, while τ' referrs to the variant where ties are not considered.

Method	**Pariwise Accuracy**	**Avg. τ**	**Avg. $\|\tau\|$**	**Avg. τ'**	**Avg. $\|\tau'\|$**
MILP reproduced	70.0%	0.58	0.66	0.44	0.56
MILP with synonymy reproduced	65.4%	0.43	0.58	0.31	0.50
Markov heuristic	67.8%	0.47	0.62	0.36	0.55
Markov pairwise approximate	71.0%	0.53	0.66	0.41	0.55
Markov pairwise mixed ILP	**75.0%**	**0.63**	**0.69**	**0.50**	**0.58**
Markov MILP	70.0 %	0.52	0.64	0.40	0.52

Table 6: Main results using Bansal's patterns and those found in table 2.