1 Introduction

2 Literature Review

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 **?**), 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.

Linguistic scale is a set of words of the same grammatical category that can be ordered by their expressive strength or degree of informativeness (**?**). 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. **?**) 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 (**?**; **?**; **?**) . 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.

?) 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 excellent, 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.

3 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).

3.1 Extracting Intensity Patterns

Both **?**) and **?**) 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-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.

strong patterns extracted by Sheinman. Bansal used a similar approach but used the Google N-gram corpus (**?**) 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.

3.2 Collecting Pattern Statistics from N-grams

We used 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 $*P*$.

3.3 Extracting PPDB Adverb Patterns

While **?**) and **?**) only considered patterns that relate adjectives within a phrase, we also considered pairs of adjectives that paraphrases each other when one of them is intensified by an adverb. Qualitatively speaking, if "very good" is a paraphrase of "great", and we know that "very" intensifies the adjective following it, then we can conclude that "good" is less intense than "great". We use the paraphrase database (PPDB) (**?**) to conduct this study. The database relate english utterances of similar meaning. Section 3.3 gives a detailed account of how we infer orderings from the database. We test our assignment on manually curated and ranked adjectives clusters.

TODO: describe how the adverb pairs are filtered by their dot product wrt word embeddings. Reference Veronica's work.

4 Problem Formulation

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

4.1 Global Ranking with Two Sided Patterns

Bansal and de Melo **?**) 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 $a_1 p a_2$ 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.

4.2 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: $\{a_i p*, a_j p*, *pa_i, *pa_j\}$. 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:

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

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:

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

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:

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

where

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

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 $P_{uv} \geq 1$:

Maximize

$$\sum_{u,v \in \{1,..,N\}} P_{uv} \cdot s_{uv} + P_{vu} \cdot (1 - s_{vu})$$

s.t

$$(1 - s_{uv}) + (1 - s_{vw}) \geq (1 - s_{uw}),$$

$$\forall u, v, w \in \{1, ..., N\}.$$

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."

4.3 Ranking using PPDB Graph.

This section describes the results of of ranking adjectives based on the PPDB graph, and explores various means of combining N-gram based data with paraphrases from the PPDB graph. We test the results on two sets of ranked clusters, those provided by Mohit Bansal, and a new hand curated set of 76 clusters.

5 Ranking Using Local and Global Graph Structure

6 Introduction and Notation.

In this chapter we explore a variety of ways of measuring relative intensities of adjectives using data from the PPDB graph. First, we explore the possiblity of ranking adjectives based only on local comparisons found in corpus and the PPDB graph. Next we consider incorporating the using local connectivity heuristic of

adjectives in cases where no direct edges are observed. Finally, we generalize this heuristic by comparing the personalized page rank of adjectives with respect to each other.

Recall the PPDB graph is a set of paraphrases over adjectives, where the vertices are adjectives and the edges are adverbs. In concert terms, the PPDB corpus reveals that the phrase "very good" is a paraphrase of the word "great", then we place a directed edge from the vertex "good" to the vertex "great". Furthermore, adverbs such as "very" and "extremely" intensify the "adjectives" they modify, while words such as "somewhat" and "kind of" deintensify said adjectives. However while poring over the data we found that [the overwhelming majority of the adjectives] are intensifiers. Moreover the adjectives modified by deintensifying adverbs are often paraphrases of themselves, or even weaker words. Thus we make the simplifying assumption that all adverbs are intensifieres, and if there are more edges from "good" to "great", then we suspect that "good" is weaker than "great."

Now we introduce some notation to formalize the above intution. The raw PPDB data forms a multi-directed graph $\widetilde{\boldsymbol{G}}$, where the vertices $\widetilde{\boldsymbol{E}}$ are adjectives, and the edges $\widetilde{\boldsymbol{E}}$ are adverbs. Each vertex will be denoted by s, t or r, while adverbs are denoted with center dot $\cdot$. One edge from s to t is denoted $(s, t; \cdot)$, note since $\widetilde{\boldsymbol{G}}$ has multi-directed, we could have multiple such edges. We can construct a directed graph from this mutli-directed graph to a directed graph by assigning some weight γ to each edge. The directed graph is named $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, vertices in this graph will also be labeled as s and t, while $(s, t; \gamma)$ denotes a directed edge from s to t with edge value γ.

7 Measure One

[Since assigning the proper value to each edge will be the basis of all our methods, we will dwell on this topic for the next few pages. We also need to construct a stochastic matrix, so all the values need to normalize to one. Need to add a sentence here jusitfying why the edge weight need to be normalized.] There are a mutlitude of issues with regard to the relative meaning of adjctives that need to be resolved in order to assign values. We leave all complications for discussion in the next section and focus on constructing the simplest baseline here. The simplest way forward is to measure outdegree of each vertex, the value γ_{st} of an edge from s to t is:

$$\gamma_{st} = \begin{cases} \frac{|(s,t;\cdot)|}{\sum_{x\in\tilde{\boldsymbol{V}}}|(s,x;\cdot)|} & (s,t,\cdot)\in\widetilde{\boldsymbol{E}}, \\ 0 & (s,x,;\cdot)\notin\widetilde{\boldsymbol{E}}, \text{for every vertex } x \end{cases} \tag{1}$$

This measure naturally leads to the following defintion of "greater than". Since the edges are intensifiers, an adjective with more outgoing edges is "intensfied" more often in common speech than those with a smaller outdegree, therefore it is weaker. Now we make this inutition precicise.

Definition 7.1. Given two adjectives s and t, we say s is less intense than t under (1)'s assignment of γ, written $s <_\gamma t$ if $\gamma_{st} > \gamma_{ts}$. Given three adjectives s, t and r, we decide $s <_\gamma t <_\gamma r$ if the following three conditions are satisfied:

$$\gamma_{st} > \gamma_{ts}$$

$$\gamma_{sr} > \gamma_{rs}$$

$$\gamma_{rt} > \gamma_{tr}.$$

If there is data for each of the three above conditions and they are consistent, then we are done. However two common problems surface in practicality: there is insufficient data for at least one of the conditions, and/or there is contradictory data. There are two ways to tackle the data sparsity problem: directly approximating the probability of a missing edge, or computing the probability arriving at t from s in a random walk. We focus on the first approach for now.

Since γ is a function of the underlying multi-directed graph $\widetilde{\boldsymbol{G}}$, we determine γ by computing the expected number edges between s and t in $\widetilde{\boldsymbol{G}}$ a priori. We assume the placement of edges from s to t is independent of the placement of edges from t to s. Now define X_{st} as a random variable ranging over a finite subset of the natural numbers $X_{st} = \{0, 1, ..., K\}$ for some large K (i.e., the size of the vocabulary) so that if $X_{st} = k$ then then are k edges from s to t. The expected value of X_{st} is:

$$\mathbb{E}[X_{st}] = \sum_{k \geq 0} Pr[X_{st} = k] \cdot k, \tag{2}$$

where the probability $Pr[X_{st} = k]$ is with respect to ??????

8 Measure Two

9 Measure Three

10 ILP

11 Personalized Page Rank

12 Logistic Regression

13 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.