Gradable Adjective Embedding for Commonsense Knowledge

Kyungjae Lee[1], Hyunsouk Cho[2], and Seung-won Hwang[1(✉)]

[1] Yonsei University, Seoul, South Korea
{lkj0509,seungwonh}@yonsei.ac.kr
[2] POSTECH, Pohang, South Korea
prory@postech.ac.kr

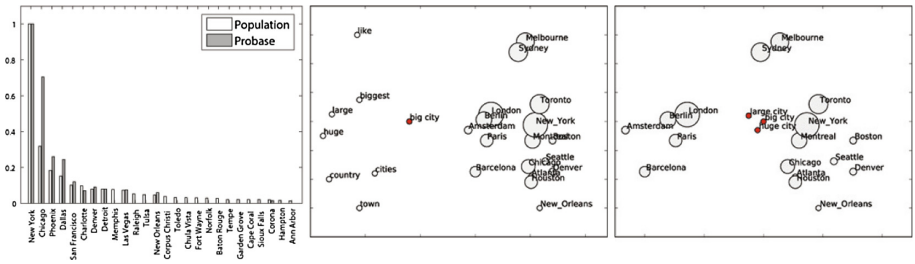
Abstract. Adjective understanding is crucial for answering qualitative or subjective questions, such as "is New York a big city", yet not as sufficiently studied as answering factoid questions. Our goal is to project adjectives in the continuous distributional space, which enables to answer not only the qualitative question example above, but also comparative ones, such as "is New York bigger than San Francisco?". As a basis, we build on the probability P(New York—big city) and P(Boston—big city) observed in Hearst patterns from a large Web corpus (as captured in a probabilistic knowledge base such as Probase). From this base model, we observe that this probability well predicts the graded score of adjective, but only for "head entities" with sufficient observations. However, the observation of a city is scattered to many adjectives – Cities are described with 194 adjectives in Probase, and, on average, only 2% of cities are sufficiently observed in adjective-modified concepts. Our goal is to train a distributional model such that any entity can be associated to any adjective by its distance from the vector of 'big city' concept. To overcome sparsity, we learn highly synonymous adjectives, such as big and huge cities, to improve prediction accuracy. We validate our finding with real-word knowledge bases.

Keywords: Adjective understanding · Commonsense knowledge · Word embedding

1 Introduction

In recent years, database and search engines have shown the effectiveness in answering quantitative questions on entities, such as "what is the population of New York". However, they are still limited in answering qualitative or subjective questions, often represented in adjective, such as "is New York a big city?" or "is New York bigger than San Francisco". This gets even harder for more subjective adjectives such as "is New York beautiful?". Adjectives, by modifying or elaborating the meaning of other words, are studied in linguistics [6] to play important roles in determining the semantic orientation of attributes, but existing computational approaches have the following limitations.

J. Kim et al. (Eds.): PAKDD 2017, Part II, LNAI 10235, pp. 814–827, 2017.
DOI: 10.1007/978-3-319-57529-2_63



(a) KB baseline (Probase) (b) DS baseline (GloVe) (c) Ours

Fig. 1. The relation between population and the score of "big city". In (b) and (c), the size of circle is proportional to the population of the city

Existing work focuses on mining textual patterns to identify if 'New York' is frequently observed with 'big city' in Hearst patterns, like 'big city such as New York' and 'New York is a big city', defining an is. A relationship between New York and big city. Specifically, Probase [19] knowledge base (KB) captures P(New York|big city) from a large web corpus, which we adopt as **KB baseline**. However, in this KB, concept *city* is modified by 194 adjectives, such that textual observations of New York are scattered over these adjective-modified concepts. Such scattering makes lesser known, or tail entities, to be scarcely observed especially in adjective-modified concepts, which we call a **observation sparsity** problem – if Urbana is not observed in the 'big city' pattern, does this mean it is not big or simply unobserved?

Trummer et al. [17] alleviate this problem by extending observations to include not only positive isA patterns, but also negative isA patterns such as 'Urbana is not a big city'. They use a provided threshold to map the given entity and adjective pair, into positive state or negative state (binary condition). However, this still cannot handle the sparsity of entities observed in neither polarity. Iwanari et al. [9] later generalize the binary classification into an ordering, using textual patterns as evidences.

We summarize the limitations as follow:

- **Observation sparsity:** As New York can be associated with virtually infinite adjectives, only few head entities are sufficiently observed in adjective-modified concepts. (For example, New York is observed as a big city, but may not be observed as a large city).
- **Human intervention:** Existing work requires human intervention to decide a score threshold or provide human-generated ordering as training data. Our goal is to give a graded score without human intervention, to not only classify whether it is big, but also to compare how big it is with respect to another entity.

Our first goal is thus to overcome **observation sparsity**, by answering "Is New York a big city?" even when no Hearst co-occurrence pattern is observed. A naive solution is adopting an existing distributed word representation

technique, of using a big corpus as input and, by unsupervised learning, distributing words that have similar meaning in the near continuous space. In Skip-gram model [13], word is represented into a vector to well predict context words defined as a surrounding slide window. Recently, GloVe [14] trains a distributional space combining the local and global model, which we adopt as **DS baseline**. This model relaxes the sparsity by not being restricted to Hearst patterns.

Our second goal is to train a graded score without human supervision. To illustrate current limitations, Fig. 1(a) and (b) show KB and DS baseline results for ordering cities to answer questions such as "Is X a big city?". KB baseline successfully grades the degree of "big city" (the bigger circle suggests higher population), but includes only a few head entities actually observed in Fig. 1(a). Meanwhile, though DS baseline overcomes the observation sparsity in Fig. 1(b) by placing all cities in the space, close vectors to 'big city' are not necessarily cities with higher population. Quantifying this failure requires a linear ordering of all cities, which requires costly human-generated labels.

We combine the strength of the two models. First, we build on sparse but highly precise Hearst patterns for training distributed word space. As a result, we obtain Fig. 1(c) where the distance from big city preserves the correlation with popularity. Meanwhile, DS baseline has higher recall but lower precision by treating all co-occurrences as equal: Highly frequent co-occurrence of 'big city' may include noisy words such as 'where', 'like', 'small'. Second, we capture the distributed similarity between adjective vectors. For example, as shown in Fig. 1(c), big and huge cities are nearly synonymous, such that scattered observations from two concepts can be combined to enhance the correlation. In other words, we can consider the distance to either vector (or the combination of the two) to predict adjective grade more robustly.

We quantify the improved performance by comparing with a total order generated by attributes (for objective adjectives), or by a total ordering generated by textual patterns (for subjective adjectives), as [17] confirms the quality of such ordering. This enables to include up to 250 concepts and 500K entities in evaluation.

2 Related Work

We categorize existing work for adjective understanding into implicit and explicit modeling. Lastly, we describe how our work complements both approaches, and describes other related attribute-related tasks.

2.1 Explicit Model

This approach considers textual patterns as explicit representation, to train a graded adjective score. Probase [19] considers Hearst patterns to extract isA relationship between concept and the given entity, observed from billions of web documents. For our purpose of adjective understanding, we can consider

Probase score for adjective-modified concepts, which we adopt as **KB baseline**. Alternatively, Trummer et al. [17] consider both positive and negative isA patterns, such as 'New York is a big city' and 'Urbana is not a big city', to train a binary classifier given the ratio of positive and negative statements. Iwanari et al. [9] use four textual patterns for finding various evidence between adjective and concept, aggregated into an ordering trained from supervised methods. This ordering is evaluated against human-generated ordering, which limits the scalability of evaluation. Our contribution is establishing Probase probability as an ordering proxy, evaluating against data attributes (for objective adjective) and missing probability (for subjective). WebChild Knowledge-Base [16] associates entity with adjectives for fine-grained relations like hasShape, hasTaste, evokesEmotion, etc.

The strength of explicit model in general is its high precision, but its weakness is missing observation. However, as there are virtually infinite combinations of adjective with concept, observations for adjective-modified concepts are typically scarce, especially for lesser known entities, for which we cannot predict the score.

2.2 Implicit Model

Meanwhile, implicit approaches leverage a neural network model and large corpus data to model latent semantic similarity between entities. For example, the continuous bag-of-words model (CBOW) and the skip-gram model [12,13] approaches predict semantic similarity between New York and Chicago based on the similarity of surrounding words, such as mayor, city, population, etc. In this space, the distance or similarity between every word can be calculated (or, achieves high recall) even if the two words did not co-occur in sentence, and the similarity will be high for two words with similar meaning. This helps infer Chicago as a big city, even when it is not explicitly observed in the Hearst pattern of "big city such as Chicago", unlike New York being frequently observed.

Similarly, LSA [5] predicts two entities being similar, based on word co-occurrence matrix. This model transforms a large co-occurrence matrix to low dimensional vectors using a dimensional reduction technique. More recently, GloVe [14] combines the strength of LSA and Skip-gram to train words into the distributed space (DS), which we adopt as **DS baseline**. Huang et al. [8] similarly predict similarity between query and document through deep learning, but this line of approach shares a common weakness of compromising precision for the increased recall.

Our goal is to increase recall without compromising precision. We thus use high precision signals from explicit model to train a distributed entity space, then infer missing scores based on its similarity with (possibly multiple synonymous) adjectives.

2.3 Joint Model and Other Attribute Work

Existing joint approach of combining implicit and explicit models can be categorized into two directions: First, we can use explicit model as a supervision to

train word embedding, such as syntactic or lexical knowledge [2,15] to improve the quality of word embedding. Second, explicit knowledge can be projected onto an embedding space [3,11,18], to enable the inference between relations. We take the advantage of both approaches, by using explicit probability as supervision for high-quality embedding, while projecting concepts in the space to enable the inference of concept-concept or concept-entity similarity.

Our work is also related to attribute understanding, as adjective is often viewed as a qualitative and subjective attributes describing the concept. First, to understand a likely set of attributes describing the concept, [10] mines "the [attribute] of [concept]" patterns. Proposed method derives attributes for millions of concepts and predicts the score of the attributes with regard to the corresponding concepts. Second, to understand similar attributes, [7] discusses how to automatically discover attribute synonyms to integrate hundreds of web tables describing the same concept.

More recently, instead of textual data, several images of objects are used for inferring the size or to predict whether the object is relatively big or small [1]. This work can capture graded property of size and complement our work, for finding 'big animal' that can be captured in the photo, but not 'big city' which cannot be photographed.

3 Proposed Model

This section first overviews existing approaches for quantifying the graded score of the given entity for adjective-modified concepts. We then propose our approach combining the strength of the two existing models.

3.1 Preliminary

Explicit Model. Probase [19] used a pattern-based method to estimate the probability between the entity and its concept from billions of Web pages. We selected only the adjective-modified concepts among various concepts in Probase and used the probability as our score. The probability between concept and entity was calculated by counting how frequently the pair of two word are found in corpus, and can be defined as:

$$P(e|c) = \frac{n(e,c)}{\sum_{e' \in E(c)} n(e',c)} \tag{1}$$

where e, c are respectively the entity and adjective-modified concept, $E(c)$ is the set of sub-entity of the adjective-modified concept c and $n(e,c)$ is the number of times (e,c) discovered by Hearst pattern. In Probase data, when an adjective-modified concept "big city" is given, the probability renders a correct size ordering, such as Chicago > London > Dublin > Washington DC, with probability 3.82%, 3.58%, 1.42%, and 0.04% respectively. Though this signal is highly precise, their coverage is limited – only 304 cities in USA (40.1%) are

observed in the Heart patterns with 'big city', though their probability score does meaningfully correlate with actual population with correlation score 0.75. However, this does not cover the rest 60% of big cities of comparable population. It is thus difficult to decide whether the unseen city is not big or simply unobserved.

Implicit Model. GloVe constructs a word embedding by using word co-occurrence data. This model trained co-occurred word vector as following equation.

$$F(w_i, \tilde{w_k}) = exp({w_i}^T \tilde{w_k}) = P(j|i) = \frac{n(i,k)}{n(i)} \tag{2}$$

where w_i is vector of word i, $\tilde{w_k}$ is separate vector of context word j, and $P(j|i)$ is the conditional probability that word j appear in the context of word i. F denotes a function that encode two vectors to real value and is used as exponential function in this model.

A naive adoption of implicit model is to train a Glove embedding and use the distance of words from the adjective-modified concept, such as 'big city. Such a naive adoption has two limitations. First, co-occurrence is more prominent with non-entity words, such as "like", "where" and "small", compared to which co-occurrence with city entities forms a long tail. This would work as a noise in generating a robust ordering among the city entities. Second, eliminating non-entity words in the space modeling cannot solve the problem either, as entity co-occurrence may bear different meanings as well. As [17] pointed out, co-occurrence of 'New York is a big city' and 'Urbana is not a big city' reflects the opposite meaning.

We discuss the joint modeling overcoming the limitations of the two models.

3.2 Embeddings for Adjectives

In semantic space represented as vectors, the distance or similarity between every word can be calculated even when two words do not co-occur in corpus. However, as training with simple co-occurrence is too noisy, we use Probase probability into the vector cosine distance in the range of -1 to 1. Therefore, we propose a model that uses word embedding and cosine distance to overcome sparsity and binary classification problem.

Loss Function for Concept. The proposed model trains adjective-modified concept into semantic word space by applying the scores to cosine similarity from the entity vector, instead of Glove model using co-occurrence. Our objective is thus to find the vector of adjective-modified concept satisfying the following condition.

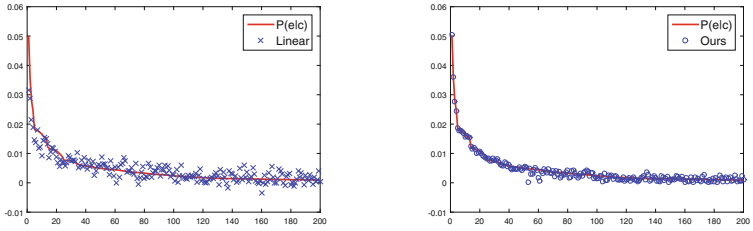
$$F(v_e, v_c) = P(e|c) \tag{3}$$

where v_c, $v_e \in \mathbf{R^d}$ are the vectors of adjective-modified concept and its entity respectively. A simple way to obtain F is by inner product:

$$F(v_e, v_c) = v_c \cdot v_e + b_e = P(e|c) \tag{4}$$

where b_e is bias of entity. As we normalize entity vectors to have size 1, this corresponds to the cosine similarity of v_c and v_e, being proportional to the probability $P(e|c)$. Suppose P(New York|big city) is higher than P(Boston|big city). Then we want to train the vector "big city" to be located closer to the vector, "New York" than "Boston". F by inner product is the same as *linear regression model*. $P(e_{1:n}|c)$ are dependent variables of (n by 1), $v_{e_{1:n}}$ are independent variables of (n by d) and v_c is intercept of (d by 1), where n is the number of data, d is the dimension of vector.

However, as motivated in Fig. 2, the frequency is showing a power-law distribution, such that F cannot fit the frequency very well. We show the errors in Fig. 2(left), contrasting with how we can improve in the right figure.



(a) Linear Regression (b) Ours

Fig. 2. Error comparison of the two models

More specifically, we modify the equation such that the inner product of two words in Eq. 2 equals to the "logarithm" of co-occurrence frequency showing Fig. 2. In other words, we can train "exponential" of the inner product to refer to the co-occurrence frequency as we reformulate as below:

$$F'(v_{e_k}, v_c) = \frac{exp(v_e \cdot v_c)}{\sum_{e_k \in E(c)} exp(v_{e_k} \cdot v_c)} = P(e|c) \tag{5}$$

where $E(c)$ is the entity set in concept c.

In Eq. 5, as the denominator is constant, $P(v_e|v_c)$ is proportional to $exp(v_e \cdot v_e)$. As a result, the entity vector placed closer to 'big city' can be bigger. For satisfying Eq. 5, the loss function of proposed method is:

$$\mathcal{L}(c) = \sum_{e_k \in E(c)} \left(P(e_k|c) - F'(v_{e_k}, v_c) \right)^2 \tag{6}$$

Global Loss Function. To optimize loss function for all adjective-modified concepts, a simple approach is minimizing $\sum_{t=i}^{T} \mathcal{L}(c_i)$, where T is the size of whole adjective-modified concept. This function considers only positively labeled data, as entities with high $P(e|c)$ to adjective-modified concept c. However, due

to the limitation of explicit model, it is unclear whether unlabeled data $e' \notin E(c)$ is missing because it is a negative evidence or simply unobserved.

To apply negative evidence, a naive method randomly samples some unlabeled data as negative data. However, it may lead to false positive of selecting unobserved big city as a negative label or insignificant effects by extracting irrelevant data. To alleviate these problems, we firstly select entities which are included in noun concept out of the adjective, but excluded in the adjective-modified concept. For example, "Urbana" is included in "city" but excluded in "big city". And secondly, we weighted entities which likely to be more negative. Our hypothesis is that, those entities that are frequently observed with city, but not particularly with big city, are more likely not to be mentioned because it is a negative evidence. Based on this observation, we define our global loss function to consider the distance with negatively unlabeled data. To avoid false positive, we use a weighted function. Our global loss function is:

$$Loss = \sum_{t=i}^{T} \left(\mathcal{L}(c_i) + \sum_{e'_k \in N(c_i)} \frac{\log(n(e'_k))}{\log(\max n(e'))} F'(v_{e'_k}, v_{c_i})^2 \right) \tag{7}$$

where $N(c_i)$ is the sampled set of unlabeled entities which are excluded in adjective-modified concept c_i, but included in noun concept out of the adjective. And $n(e')$ is the sum of the frequency of entity e'. Through this approach, we can enhances the accuracy, as our empirical results confirm in Table 4 (precision improves by 6.9%).

3.3 Finding Adjective Synonym

This section reports how distributed space can be used to detect semantic relationship between adjectives. In Fig. 1(c), "big city" is placed near "large city", "huge city". We can observe that closest adjectives are all highly semantically related. This suggests that using highly related adjectives as a cluster can aggregate scattered observations of "big company" to "large company" or "huge company".

We can aggregate the closest pair at each iteration, until they converge to synonym clusters, by adopting a bottom-up agglomerative hierarchical clustering method [4]. Specifically, we compute a pairwise distance matrix using cosine similarity in word embedding and use it for clustering: We can continue iterative merges until the number of adjectives in one group is 4 or less.

Then, we combine the adjective set in cluster, *i.e.* (big, large, huge city), by using the average score of synonyms, instead of the score of one adjective. As shown in Table 4, we can show whether combining the statistical evidences from similar adjectives can enhance the quality of the graded score prediction. This average score indeed enhances the accuracy, as our empirical results confirm in Table 4 (precision improves by 19.2%).

4 Experiments

This section is organized to answer the following research questions respectively.

- RQ1: Some adjective can be used when people want to express objective properties such as the size of the country. Therefore, we select some obvious qualitative adjective and check the correlation with objective statistics to show how our model captures such correlation.
- RQ2: Meanwhile, there exist non-measurable or subjective adjectives such as great, valuable, or beautiful. We evaluate our model for these properties by using human-made gold standard ordering.
- RQ3: We validate whether our model overcomes the limitation of explicit models by extending prediction of $P(e|c)$ to unseen objects.

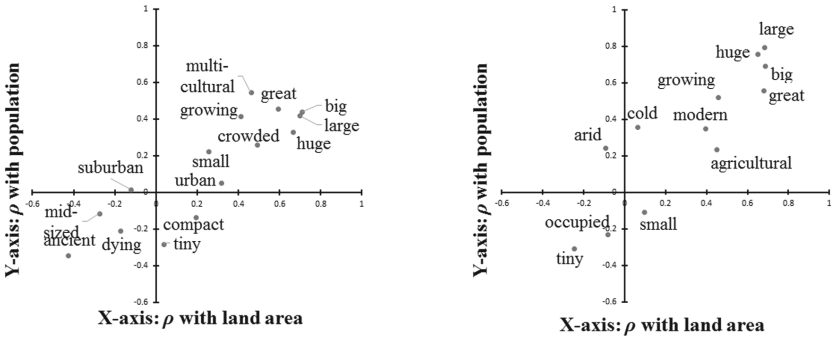
4.1 RQ1: Interpreting Qualitative Adjective with Statistics

Some adjectives naturally correlate with objective statistics, such as big city with statistics of population or area. We first demonstrate whether such correlation confirms commonsense understanding of humans. We generalize our observation to qualitative adjectives in Table 1, by using 8 field statistics. In these fields, we show Spearman correlation between statistics and the graded scores calculated by cosine similarity in word embedding.

In Table 1, we observe $P(e|c)$ in KB baseline reflects human-perceived correlation, but covers only a limited number of data. For example, KB baseline grades "big city" only for 186 cities, but our model scores them for 278 cities. Our model obtains high coverage, calculating the similarity of the entities that cannot be extracted by specific pattern. The last column shows that our model expands the coverage while preserving correlation.

Table 1. Spearman's rho between the graded score and the statistics

Adjective-modified concept	Statistics type	KB baseline		Ours	
		Correlation	# of data	Correlation	# of data
Big city	Population	0.705	186	0.706	278 (149%)
	Land area	0.460	186	0.446	278 (149%)
Expensive city	Big mac Index	0.630	54	0.648	70 (130%)
	Cost of living	0.571	282	0.508	444 (157%)
Large country	Population	0.651	119	0.741	191 (161%)
	Land area	0.799	119	0.803	191 (161%)
Rich country	GDP	0.690	120	0.690	169 (141%)
	PPP GDP	0.717	120	0.708	169 (141%)



(a) City (b) Country

Fig. 3. Correlations between adjectives and population/land area

We observed that the correlations between adjective and statistics are different depending on the combined concepts. As shown in Fig. 3, "large country" is more highly correlated with population than land area, but "large city" is more highly correlated with population than land area. The opposite meaning of "large" is "ancient" or "dying" in city, but "tiny" in country. "dying" and "ancient" are rarely used in country, and human tend to represent "small town" for negative correlation word for population and land area not "tiny city". The use of "large city" correlates more with population than land area, while "large country" correlates more with land area. This also confirms the human perception of considering countries such as Russia, China, or the US with large area as big countries, while considering metropolis with high population as big cities.

4.2 RQ2: Comparing Correlation with Human-Made Gold-Standard

To evaluate our model in terms of correlation, we adopt the gold standard orderings made by human. Iwanari et al. [9] release evaluation dataset including 35 adjective-modified concepts and average 7 entities per each concept. They asked multiple volunteers to order some entities set on attribute intensity expressed by adjective. Then, they pick the ordering that achieved the best average Spearman's correlation and use the ordering as gold standard. However, unlike our English dataset, the model was built on Japanese corpus and evaluation set. Because the domain of data we use is English, we excluded 16 specific concepts related to Japanese, such as cartoon, alcohol, temple, corner store, and town. Finally, we use ordering between 19 concepts and 134 entities for comparing our model. Additionally, Iwanari et al. [9] translated the concept and adjective words into English. However, the translated words are in less general form, we changed the words to synonyms that are more frequently used. For example, we chose the word "intelligent animal", instead of "clever mammal" in dataset.

The experimental results are listed in Table 2. SVM and SVR refer to the methods proposed by Iwanari et al. [9]. KB baseline refers to the pattern-based

Table 2. Spearman's rho against gold-standard ordering

Adj.concept	Human	KB baseline (coverage)	SVM	SVR	Ours
Beautiful plant	0.767	0.866 (37.5%)	0.357	0.167	**0.381**
Valuable gemstone	0.682	0.782 (87.5%)	0.524	0.548	**0.643**
Popular sport	0.422	0.290 (75.0%)	**0.381**	−0.095	0.238
Intelligent animal	0.598	0.400 (66.7%)	0.143	0.029	**0.600**
Large animal	1.000	0.500 (50.0%)	0.771	**0.886**	0.600
Great food	0.639	0.058 (75.0%)	**0.607**	0.464	0.143
Beautiful instrument	0.583	0.257 (75.0%)	0.310	0.238	**0.548**
Easy language	0.845	0.750 (87.5%)	0.619	0.643	**0.667**
Slow language	0.840	0.100 (62.5%)	**0.381**	0.238	−0.167
Lovely animal	0.806	1.000 (37.5%)	0.548	0.595	**0.738**
Great vegetable	0.462	0.696 (75.0%)	**0.524**	0.476	0.429
Sweet fruit	0.729	0.783 (71.4%)	0.607	0.607	**0.821**
Great tool	0.772	0.300 (71.4%)	0.393	**0.500**	0.464
Good protein	0.662	0.900 (71.4%)	0.143	−0.286	**0.964**
Safe country	0.804	0.300 (100%)	−0.200	0.000	**0.500**
Warm country	0.961	0.866 (60.0%)	0.700	0.700	**1.000**
Well-known brand	0.659	0.743 (87.5%)	0.619	0.286	**0.900**
Nice browser	0.856	0.600 (80.0%)	−0.600	−0.600	**0.429**
Safe city	0.655	0.378 (100%)	0.357	0.250	**0.762**
Average	0.723	0.556 (72.2%)	0.378	0.297	**0.561**

method by the probability in Probase. While KB baseline has a coverage of only 72.2%, our model has not only 100% coverage, but also preserves precision.

We see that the correlation between our score and the gold standard ordering is less than 0.4 for "popular sport", "beautiful plant", "popular sport", "great food", and "slow language". The reasons for this result are that the coverage at extracted positive evidence is low or human's agreement is inconsistent due to its subjective property.

4.3 RQ3: Generalizing Beyond Implicit and Explicit Models

In this section, we evaluate further on how we predict $P(e|c)$ for unseen pairs during the training. Table 3 shows how we expand the observation made for four adjective-modified concepts into 250 concepts.

More specifically, to validate our model for unobserved entities, we set some $P(e|c)$ to test set and estimate that probability. By Eq. 5, the probability $P(e|c)$ and cosine similarity between e and c are monotonically increasing. Therefore, we evaluate the Spearman correlation of the cosine similarity and $P(e|c)$ that were not used in the training.

Table 3. Datasets

	Concept	Adjective-modified concept	Entity–Adj.concept pair
# of data	37	250	498,007
Example	City, Country, Company, Sport, Movie	Big city, Rich country, Great sport, Funny movie, Big company	Big city-New York, Great sport-Tennis, Big company-Apple

Table 4. Experimental results

Model	ρ
KB (5)	0.434
KB (10)	0.477
DS	0.484
KB+DS (5)	0.454
KB+DS (10)	0.461
Ours (Eq4)	0.469
Ours (Eq6)	0.535
Ours (Eq6+7)	0.572
Ours (Eq6+8)	0.641
Ours (Eq6+7+8)	**0.682**

For experiment, we split the entities which have the probability $P(e|c)$ to 9/10 training set and 1/10 testing set. Because the distribution of $P(e|c)$ is skewed, random sampling selects mostly tail entities with low probability similar to each other. Such sampling is inappropriate for comparing the correlation with the actual and predicted ranking, as it contains mostly tail entities with tied ranks. We thus sample more on head entities using stratified sampling, dividing sample size n into 5 section by rank and select $\frac{n}{2^i}$ data from the highest rank ($i = 1$) to lowest ($i = 5$).

We compare with KB and DS baselines, and consider its combination as well. It is to show how each component technique we proposed contribute to overall performance.

Baselines:

- **KB baseline:** KB baseline itself cannot be used for estimating missing $P(e|c)$, but we can extend by averaging the probability of the nearest 5 or 10 concepts, which we denote as $KB(5)$ and $KB(10)$ respectively.
- **DS baseline:** In DS baseline, we estimate $P(e|c)$ by averaging the vector of adjective and noun and computing the distance to this vector.
- **KB + DS baseline:** In KB+DS baseline, we estimate $P(e|c)$ by averaging the probability of 5 and 10 closest entities e' in the word embedding, denoted as $KB + DS(5)$ and $KB + DS(10)$.

Our model outperforms all these baselines. To show how we each equation contributes to the overall performance, we denote the complete model as Ours (Eq. 6+7+8) in the last line, which we compare with our model applying only some of such equations.

5 Conclusions

This paper studied the problem of understanding adjective by predicting a graded score for the given entity and adjective pair. Specifically, we train a distributed space to reflect Probase probability as distance. Semantic similarity with unseen objects is then used to predict missing Probase probability. Semantic similarity between adjectives contributes to enhance recall by collapsing the

scattered observations of an entity with synonymous adjectives. Our extensive analysis using real-life data validates that we can predict adjective for unseen entity with comparable quality to seen ones, and thus improves the coverage to all adjective entity pairs.

Acknowledgment. This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. B0101-16-0307, Basic Software Research in Human-level Lifelong Machine Learning (Machine Learning Center)).

References

1. Bagherinezhad, H., Hajishirzi, H., Choi, Y., Farhadi, A.: Are elephants bigger than butterflies? Reasoning about sizes of objects. arXiv preprint arXiv:1602.00753 (2016)
2. Bian, J., Gao, B., Liu, T.-Y.: Knowledge-powered deep learning for word embedding. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8724, pp. 132–148. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44848-9_9
3. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: Conference on Artificial Intelligence (2011)
4. Day, W.H., Edelsbrunner, H.: Efficient algorithms for agglomerative hierarchical clustering methods. J. Classif. **1**(1), 7–24 (1984)
5. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391 (1990)
6. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics (1997)
7. He, Y., Chakrabarti, K., Cheng, T., Tylenda, T.: Automatic discovery of attribute synonyms using query logs and table corpora. In: International World Wide Web Conferences Steering Committee, WWW (2016)
8. Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: CIKML. ACM (2013)
9. Iwanari, T., Yoshinaga, N., Kaji, N., Nishina, T., Toyoda, M., Kitsuregawa, M.: Ordering concepts based on common attribute intensity. In: IJCAI (2016)
10. Lee, T., Wang, Z., Wang, H., Hwang, S.-W.: Attribute extraction and scoring: a probabilistic approach. In: ICDE. IEEE (2013)
11. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: AAAI (2015)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
13. Mikolov, T., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems (2013)
14. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP (2014)
15. Rothe, S., Schütze, H.: Autoextend: extending word embeddings to embeddings for synsets and lexemes. arXiv preprint arXiv:1507.01127 (2015)

16. Tandon, N., de Melo, G., Suchanek, F., Weikum, G.: Webchild: harvesting and organizing commonsense knowledge from the web. In: WSDM. ACM (2014)
17. Trummer, I., Halevy, A., Lee, H., Sarawagi, S., Gupta, R.: Mining subjective properties on the web. In: SIGMOD. ACM (2015)
18. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: AAAI. Citeseer (2014)
19. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probase: a probabilistic taxonomy for text understanding. In: SIGMOD. ACM (2012)