Modeling the Ordering of English Adjectives using Collaborative Filtering

Sagar Indurkhya
Massachusetts Institute of Technology
32 Vassar St.
Cambridge, MA 02139
indurks@mit.edu

Abstract

We present a novel procedure for acquiring productive knowledge of restrictions on adjective ordering from counts of adjective-bigrams, which are readily obtainable from natural language corpora. The procedure uses a model-based Collaborative Filtering (CF) algorithm, and is the first computational model of adjective-ordering to do so. We consider two widely-used model-based CF algorithms, Singular Value Decomposition and Non-negative Matrix Factorization. We evaluated the procedure by first training the underlying CF model on subsets of the largest publicly available dataset of English adjective-bigrams, the Google Books NGram database, and then measuring the model's capacity to predict the ordering of (unseen) pairs of adjectives. Our results show that both CF models exhibit good performance on the task of predicting adjective ordering. Moreover, the CF models consistently outperform a baseline model that is grounded in a ranking of adjectives intended to align with a (linear) hierarchy of adjective-classes, suggesting that CF models make use adjective-ordering data that does not neatly fit into (proposed) hierarchies of adjective-classes.

1 Introduction

Linguists have long been observing, studying and characterizing restrictions on the ordering of adjectives – e.g. native English speakers will say *"the big red ball"* and not *"the red big ball"* (Rizzi and Cinque, 2016; Trotzke and Wittenberg, 2019). Typically, linguists have grouped adjectives into semantic classes over which an ordering is proposed - e.g. (Goyvaerts, 1968; Vendler, 1968; Dixon, 1982; Shaw and Hatzivassiloglou, 1999; Cinque, 1994) propose an ordering over the following classes of adjectives (where $A{>}B$ indicates A precedes B):

SIZE > SHAPE > AGE > COLOR > PROVENANCE

Moreover, linguists have noted recurring patterns of restrictions on adjective ordering that appear over a wide and diverse range of languages, and these patterns can be assembled together to form a cross-linguistic hierarchy of adjectives that informs the ordering of adjectives that (directly) modify a noun phrase (Sproat and Shih, 1991; LaPolla and Huang, 2004; Trainin and Shetreet, 2021).[1] Given that these *tacit* restrictions on adjective ordering appear consistently across languages, it is puzzling how, and the degree to which, learners acquire this knowledge of language from the primary linguistic data. This study addresses this puzzle by introducing a novel procedure for acquiring *productive* knowledge of restrictions on adjective ordering.

Our procedure, implemented as a working computer program, takes as input adjective-bigram statistics listed in the *Google Books NGram database*, and outputs a (learned) model of adjective-ordering preferences. Importantly, the procedure learns a model of adjective-ordering preferences that is *productive* in that it can predict that a speaker would prefer to say *"the big red ball"* and not *"the red big ball"* even if it has never seen the adjective bigrams *"big red"* and *"red big"*, so long as the input data includes other adjective bigrams that involve *"big"* and *"red"*. In this way, the model output by the procedure goes beyond the null-hypothesis of learners simply repeating back what they have heard (Bar-Sever et al., 2018). The procedure centers on a model-based Collaborative Filtering (CF) algorithm that maps adjectives-ordering data to a low-dimensional embedding space where latent relationships between adjectives are surfaced; our approach is informed by earlier work suggesting that CF can be employed to model constituent selection more broadly (Indurkhya, 2021). Our experiments show that model-based CF algorithms perform well on

[1] This led to the *cartographic enterprise*, which aims (in part) to provide a syntactic accounting, in the form of a detailed map, of the observed cross-linguistic hierarchy of adjective-classes (Scott, 2002; Laenzlinger, 2005; Cinque, 2010; Shlonsky, 2010; Cinque and Rizzi, 2012; Cinque, 2014).

the task of predicting the ordering relation (if there is one) between the two (input) adjectives. Moreover, the CF algorithms substantively outperform a baseline model that computes the relative difference in the ranking of the two (input) adjectives – this baseline model is informed by prior work that suggested that a ranking of adjectives that correlates with a (linear) hierarchy of adjective-classes can be used to determine restrictions on adjective ordering. Our experiments thus suggest that model-based CF algorithms leverage adjective-ordering data that does not neatly align with proposed hierarchies of adjective-classes.

The remainder of this study is organized as follows. We begin by discussing prior work that (computationally) modeled restrictions on adjective ordering (see §2) and review established methods for CF (see §3). Next, we walkthrough the construction of an Adjective Ordering matrix from adjective-bigram data drawn from the Google Books NGram database, and make note of cylical orderings over adjective that defy the organization of adjectives into a hierarchy (see §4). We then detail the computational experiments central to this study, and analyze the information used by the CF models we evaluated (see §5). Finally, we conclude by discussing the broader implications of our study, especially in light of the minimal assumptions our approach aims to make (see §6).

2 Prior Studies of Adjective Ordering

Previous (computational) models of adjective ordering can broadly be construed as falling into one of two categories that are distinguished by what they aim to explain. The first category of work includes empirically-grounded methods that aim to explain the distribution of adjective orderings observed in corpus data. The second category of work includes models that encode some proposed measure of adjectives and that aim to explain the proposed cross-linguistic hierarchy of adjectives (from which the ordering of adjectives can be deduced).[2] Note that this two-fold categorization of prior work is not a strict dichotomy - e.g. work aiming to explain how a cross-linguistic hierarchy is learned may involve an empirically grounded approach. Let us now examine these two categories of prior work in more detail.

[2]As (Svenonius, 2008) notes, each adjective-class in the hierarchy can be mapped to a functional head that may be incorporated into a determiner phrase.

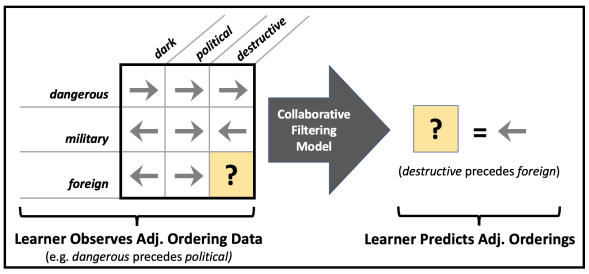

Figure 1: CF model for predicting the ordering of adjective pairs. Evidence supporting the prediction is: (i) *destructive* is similar to *dark* w.r.t. preceding *military* and following *dangerous*, and *dark* precedes *foreign*; (ii) *foreign* behaves like *military* w.r.t. preceding *political* and following *dark*, and *destructive* precedes *military*.

Prior work falling into the first category relies on corpus data from which the statistics of nouns and their adjectival pre-modifiers can be derived. Shaw and Hatzivassiloglou (1999) outline some of the standard methodologies employed, which we briefly describe here:

(a) *Direct Evidence* method: if the counts of adjective-bigrams "A B" and "B A" are c_{AB} and c_{BA} (respectively), a binomial test is used to determine whether the ratio of c_{AB} to c_{BA} differs significantly from the null-hypothesis of a 1:1 ratio – if the null-hypothesis is rejected, then we know speakers prefer "A B" over "B A" if $c_{AB} > c_{BA}$ (and vice versa). A weakness of this method is that it is *not* productive as it can only predict ordering preferences for adjective pairs that appeared as bigrams in the input corpus data.

(b) *Transitivity* method: given adjectives A, B and C, and evidence that $A > B$ and $B > C$ (perhaps determined via the binomial test outlined in the Direct Evidence method), this method will infer via transitivity that $A > C$. This method, which relates to the notion of ordering being derived from a hierarchy of adjective-classes, runs into difficulty when adjective ordering preferences inferred from the (input) corpus data violate transitivity. (See Table 1 for examples of adjective-bigram cycles.)

(c) *Clustering* method: adjectives are clustered together (e.g. via k-Nearest Neighbors) based on their ordering relation to other adjectives (with ordering determined via the binomial test as outlined in the Direct Evidence method), and the ordering of adjectives (A, B) is made by examining how other adjectives similar to A are ordered with respect to B. Similar to

memory-based CF algorithms (reviewed in §3), this approach suffers from sparsity of adjective-bigrams in the (input) corpus data, with many adjective pairings appearing only once in the data (Malouf, 2000).[3]

Although these empirically-grounded methods achieve reasonably good performance when applied to specific or restricted domains of text (where there are fewer occurrences of polysemy), their performance declines when applied to broader corpora of text that span multiple domains.

The second category of prior work aims to account for the (cross-linguistic) hierarchy of adjectives-classes. Some work in this category introduces a metric that can be used to rank adjectives, with the ordering over a pair of adjectives determined by comparing the rank of the two adjectives. For example, Hahn et al. (2018) introduces a metric that computes the mutual information between an adjective and the nominal head it modifies, so that given a nominal head, N, adjectives having higher mutual information with N will appear closer to N; this metric was grounded in the subjectivity theory of Hill (2012) and Scontras et al. (2017), with the latter carrying out experiments showing that *property-subjectivity* is what predicts adjective ordering - i.e. adjectives that are less subjective appear closer to the (modified) nominal head.[4] Alternatively, other work takes the approach of directly inferring a hierarchy of adjective classes. For example, recent work by Leung et al. (2020) demonstrates that latent-variable models (that predict adjective ordering) can learn cross-linguistic adjective hierarchies that align with Cinque's hierarchy – their model represents adjectives using *multi-lingual* word embeddings, which can (implicitly) encode information pertaining to adjective ordering in the form of the presence or absence of (nearby) non-adjectival tokens - e.g. the non-adjectival tokens that form the context of a word (that is being embedded) may encode the semantic category the word belongs to, in turn allowing it to be placed within Cinque's hierarchy.[5]

To summarize, this study primarily falls under the first category of work, although it is informed by prior work from both categories of prior work in so far as: (i) we employ a binomial test to determine adjective ordering from bigram counts as in Shaw and Hatzivassiloglou (1999); (ii) based on the assessment of Malouf (2000), we opt to directly work with the adjective-bigram counts listed in the *Google Books NGram database*, and thus do not consider the words surrounding the adjective-bigrams (see §4); (iii) the baseline model we test the CF models (output by our procedure) against is informed by observations in (Scontras et al., 2017; Hahn et al., 2018) that adjectives can be ranked, with an adjective's ranking determining it's ordering relative to other adjectives (and with the ranking meant to correlate with the hierarchy of adjective-classes). Finally, as we discuss in §3, our decision to use model-based CF algorithms is motivated in part by the need for a *productive* model that robustly deals with *sparse* (adjective-bigram) data drawn from a single language (i.e. no use of multilingual word embeddings), builds on the ideas underlying the *memory-based* models referenced in Shaw and Hatzivassiloglou (1999), and does not make assumptions about the *transitivity* of restrictions on adjective ordering.

3 A Review of Collaborative Filtering

The Collaborative Filtering (CF) algorithms used in this study are (widely used) examples of Recommender Systems (Herlocker et al., 2004; Su and Khoshgoftaar, 2009; Lü et al., 2012; Bobadilla et al., 2013). Given a finite set of users (e.g. subscribers), a finite set of items (e.g. movies), and a *user-item rating matrix* that encodes ratings assigned by (some) users to (some) items, a Recommender System is tasked with predicting the rating a given user would assign to a given item; these predictions may then be used to enumerate a list of recommended items for the given user. This study takes the users to be the first adjective in an adjective bigram, the items to be the second adjective in an adjective bigram, and the *user-item rating matrix* to be an *adjective-ordering matrix* (detailed

[3] Malouf (2000) also notes that trying to incorporate information from the text surrounding an adjective bigram doesn't provide sufficient syntagmatic information because the left side is usually a determiner and the right hand side (typically a nominal) only worsens the data sparsity issue.

[4] Note that this approach only partly aligns with there being a *hierarchy* of adjective classes as the experiments in Scontras et al. (2017) showed that users sometimes have differing subjectivity ratings for adjectives – this aligns with our observation (detailed in §4) that there are large numbers of adjective pairings for which there is no clearly preferred ordering. (See also Scontras et al. (2019).)

[5] The experiments detailed in Leung et al. (2020) control for explicitly encoded information about adjective ordering in the text from which the word-embedding models are learned; they do not, however control for implicitly encoded information in the form of the presence or absence of non-adjectival tokens incorporated into the word-embedding model.

Adj. Type	Examples		
	Ordered Adj. Pairs	Unordered Adj. Pairs	Adj. Cycles of Length 3
all	(important, private), (abbreviated, new)	(relative, domestic), (foreign, strategic)	(permanent, unconscious, young), (acoustic, grand, old)
pure	(intriguing, new), (international, technological)	(recent, substantial), (everyday, practical)	(ethical, foreign, institutional), (elementary, historical, prospective)
noun	(independent, moral), (radical, socialist)	(overall, specific), (fundamental, common)	(aged, former, intermediate), (medium, stiff, ordinary)
verb	(certified, registered), (corresponding, free)	(dry, round), (open, parallel)	(flat, major, long), (appropriate, major, free)

Table 1: Examples for each subset of the adjective-bigram data. An ordered pair of adjectives, (a, b), indicates that a typically precedes b (as determined by the test outlined in §4). An unordered pair of adjectives, (a, b), indicates that there is no preference of either a preceding b or b preceding a. Finally, an adjective cycle, (a, b, c), indicates the presence of three ordered pairs, $\{(a, b), (b, c), (c, a)\}$, that together form a cycle (and cannot be folded into a strictly linear hierarchy of adjectives).

in §4) – then a Recommender System may be used to predict, for a given adjective, which adjectives may follow it to form an adjective bigram.

Recommender Systems are typically divided into *Content-Based (CB) recommendation algorithms* and *CF recommendation algorithms*.[6] CB recommendation algorithms make use of similiarities between the valuations of features associated with each item (or alternatively, between the valuations of features associated with each user). For example, a CB recommendation algorithm can predict whether the adjective *"red"* can follow the adjective *"large"* by comparing the syntactic and semantic features associated with *"large"* with those associated with other adjectives known to follow *"large"* (such as *"blue"*). If these semantic and syntactic features are unavailable – e.g. as in this study, in which we intentionally restrict our attention to adjective bigram statistics derived from (unannotated) corpus data – then we can instead use CF recommendation algorithms, which can simultaneously take into account: (i) similarities between users, as measured by how similarly they rate items; (ii) similarities between items, as measured by how similarly they are rated by users. Conventionally, CF algorithms are grouped into two classes, *memory-based* CF algorithms and *model-based* CF algorithms, which we will now describe in turn.

Memory-based CF algorithms operate on the assumption that the more similar two users are (with respect to the ratings they assign), the more likely they are to assign similar ratings to items. A Memory-based CF algorithm predicts the rating a user, u, would assign to an item, i, by: (i) identifying a set of users, S, that are similar to u – e.g. by computing the *k Nearest Neighbors* using a similarity measure such as the Pearson correlation coefficient; (ii) computing the predicted rating as the weighted average of ratings assigned by each $s \in S$ to i, with the weights corresponding to the degree-of-similarity of each s to u (Schafer et al., 2007). More broadly, a memory-based CF algorithm is either an instance of *user-based* collaborative filtering, in which case it operates by identifying similar users (as described above), or alternatively, *item-based* collaborative filtering, in which case it identifies similar items. See (Sarwar et al., 2001) for a review of *item-based* CF.

Despite enjoying widespread usage, memory-based CF algorithms struggle in two scenarios: first, the quality of their predictions quickly degrades when the user-item rating matrix is sparse, and second, they do not scale well (with respect to memory consumption) as the number of users and items grow (Adomavicius and Tuzhilin, 2005). These difficulties are addressed by **Model-based CF algorithms**, which center on a *learned* predictive model. Notable examples of model-based CF algorithms include *Non-negative Matrix Factorization* (NMF) and *Singular Value Decomposition* (SVD). NMF and SVD, both instances of *latent factor models*, work by factoring apart the *user-item rating matrix*, so that the user and item profiles (corresponding to the rows and columns of the *user-item rating matrix*) are embedded in a lower-dimensional space where latent relationships

[6] (Burke, 2002, 2007) detail hybrid recommender systems that fuse together aspects of CB and CF algorithms.

between users and items are more readily apparent; in this way, NMF and SVD address two weaknesses of memory-based CF algorithms, *sparsity* and *scalability*. Finally, we note that NMF and SVD yield *linear* models that have the capacity to encode a hierarchy of adjective-classes (e.g. Cinque's hierarchy).[7] For these reasons, the present study opts to use two (latent factor) model-based CF algorithms, NMF and SVD, to model adjective-ordering.

4 Constructing an Adj. Ordering Matrix

We now detail how to derive an adjective-ordering matrix (i.e. the CF-model's input) using bigram data taken from the American-English (2019) subset of the *Google Books NGram database* (for the years 1969-2019) (Michel et al., 2011; Lin et al., 2012).

To begin, we define a *word pair* to be a two-tuple of words, (x, y), such that x lexicographically precedes y. We restricted our analysis to only consider a word pair (x, y) if met three conditions:

(a) let A be the set of all adjectives appearing in the Google Books NGram database (as marked by the Part-of-Speech tagging of each ngram in the database), then $x, y \in A$;

(b) the sum of the frequencies of the bigrams "x y" and "y x" is at least 100 – this eliminates word pairs with insufficient samples for statistical hypothesis testing;

(c) let W be the set of adjectives in the WordNet database (Miller et al., 1990; Miller, 1995), and let W' be the set of adjective lemmas obtained by lemmatizing members of W – then $lemma(x) \in W'$ and $lemma(y) \in W'$.[8][9]

There were 491499 distinct word pairs that met these conditions.

Next we define an *adjective pair* as a two-tuple of lemmas, (s, t), such that s and t are both lemmas of adjectives that make up a word pair; there are 472823 distinct adjective pairs, and 10041 distinct adjective-lemmas appeared in these adjective pairs. Given an adjective pair (s, t), we define its *forward frequency*, $f(s, t)$, as the sum of the frequencies of bigrams of the form "x y" where $s = lemma(x)$ and $t = lemma(y)$.[10] Likewise, we define the *backward frequency*, $b(s, t)$, as the sum of the frequencies of bigrams of the form "y x" where $s = lemma(x)$ and $t = lemma(y)$. Finally, we define the *ratio*, $r(s, t)$, as $\frac{f(s,t)}{f(s,t)+b(s,t)}$.

We then marked each adjective pair as either being *ordered* or *unordered*. To make this determination for an adjective pair (s, t), we used a two-tailed binomial test, with probability of success $r(s, t)$, to evaluate the null hypothesis that bigrams "x y" and "y x", where $s = lemma(x)$ and $y = lemma(y)$, are equally likely to appear (because there is no significant preference in the ordering of the two adjectives, and thus the adjective pair is determined to be *unordered*). We rejected the null hypothesis (implying the adjective pair is *ordered*) if $p < \frac{0.01}{n^2}$, where $n = 10041$ is the number of distinct (lemmatized) adjectives and n^2 is the Bonferroni correction factor.[11] Of the adjective pairs, 344228 were found to be ordered and 128595 were found to be unordered. Table 1 shows examples of ordered and unordered adjective-pairs.[12]

We can now define the *adjective-ordering matrix*, Q (a 10041×10041 matrix), as follows. Let L be the set of adjectives pairs. Then for each $(s, t) \in L$, (i) if (s, t) is unordered, then $Q[s, t] = Q[t, s] = 1$; (ii) if (s, t) is ordered, then $Q[s, t] = 2$ if $r(s, t) > 0.5$, otherwise $Q[t, s] = 2$. Any entry in Q not defined above has value 0.[13]

To better understand how the (alternative) lexical categories that a polysemous adjective can appear as may impact a model's ability to learn ordering information,[14] we also produced adjective-ordering matrices for subsets of the set of adjective pairs - these subsets are labeled as follows: *all* denotes

[7] This is possible if each adjective is mapped (via it's embedding vector) to a weighted sum of adjective classes – then the learned matrix can encode information about which adjective classes can precede which other adjective classes.

[8] Checking for membership in WordNet's list of adjectives guards against potential mislabeling of Part-of-Speech tags in the Google Books NGram database, spelling errors, etc, and enables identification of adjectives that are also verbs or nouns (using WordNet's lists of nouns and verbs).

[9] The lemmatized form serves to normalize the various forms of adjectives (e.g. superlatives and comparatives). of W'. We used spaCy (v3.2.0) to lemmatize words.

[10] The frequency of a bigram is the number of times it appears in the Google Books corpus, as recorded in the Google Books Ngram database. Our measurement of bigrams frequency was case-insensitive.

[11] Bonferroni correction counteracts the propensity for false-positives arising when doing multiple comparison testing.

[12] The reader can inspect an example of an adjective pair (a, b) by going to the *google-ngrams* website (`https://books.google.com/ngrams`) and running the query "`a_ADJ b_ADJ, b_ADJ a_ADJ`" with the queried data restricted to the years 1969-2019.

[13] We use values of 2 and 1 so that the adjective-ordering matrix can serve as input to the Non-negative Matrix Factorization (NMF) CF model; the CF models ignore the 0 values.

[14] E.g. the polysemous word *"sound"* can appear as a noun or as an adjective, each having a different meaning. See also (Taylor, 2003; Baker and Baker, 2003; Falkum, 2015).

Adj. Type	Adjective-Ordering Matrix Statistics				Model Performance (AUROC)			t-Test Statistic	
	Adjs.	Adj. Pairs	Ordered Pairs	Cycles	NMF	SVD	Baseline (γ)	NMF	SVD
all	10041	472823	72.8%	970527	**0.788**	0.770	0.716	28.972	33.420
pure	6817	76228	70.3%	15661	**0.756**	0.738	0.676	5.915	26.799
noun	2688	130701	75.6%	311491	**0.767**	0.734	0.710	23.431	10.143
verb	1755	26284	71.2%	22430	**0.716**	0.689	0.662	10.426	13.654
acyclic-all	9296	382335	66.6%	0	**0.867**	0.862	0.791	45.505	46.278
acyclic-pure	4776	64843	66.3%	0	**0.813**	0.809	0.755	6.239	27.481
acyclic-noun	2450	105938	70.0%	0	**0.858**	0.853	0.795	29.530	9.344
acyclic-verb	1265	21657	65.8%	0	**0.797**	0.788	0.758	10.367	9.586

Table 2: Performance of two Collaborative Filtering models (NMF and SVD) and a baseline (γ) model on each adjective-ordering matrix that was derived from the (Google NGrams) Adjective Bigram data. Notably, the NMF model consistently has the top performance, and both Collaborative Filtering models consistently outperform the baseline model. *Adj. Type* indicates the subset of the data from which an adjective-ordering matrix was derived and whether or not adjective cycles are present. Key statistics are presented for each adjective-ordering matrices, and for each (model, dataset) pair, we report for the median-scoring model: (i) the performance (as measured by AUROC), and (ii) the Welch's t-test statistic that is used to compare the distributions of the *fraction of adjacent adjective pairs* for correctly and incorrectly classified adjective pairs (see the analysis in §5 for details).

the original set of adjective pairs; *noun* denotes the subset of *all* in which both lemmas in an adjective pair appear in the lemmatized list of nouns in WordNet; likewise, *verb* denotes the subset of *all* in which both lemmas in an adjective pair appear in the lemmatized list of verbs in WordNet; *pure* denotes the subset of *all* in which both lemmas in an adjective pair are not members of the *noun* or *verb* subsets. Table 2 presents statistics for each adjective-ordering matrix.

Finally, we note that the *ordered* adjective pairs can be modeled as a directed graph: an adjacency matrix encoding a directed graph, where nodes correspond to distinct lemmatized adjectives, may be obtained by removing the entries in Q associated with unordered adjective pairs. Upon constructing these graphs, we identified the presence of *adjective cycles*, which are cycles of edges in the graph formed by a sequence of adjective bigrams - see Table 1 for examples of these cycles, and see Table 2 for counts of adjective cycles of length three. *The presence of these adjective cycles surprised us as it seems to rebut the ordering hierarchy over adjectives proposed by the Cartographic Enterprise.* To better understand the role that the presence of these cycles may play in a model's ability to learn the ordering of adjectives, we constructed a directed acyclic graph (DAG) that was a subgraph of the directed graph derived from the adjective-ordering matrix. We did this by identifying a *feedback arc set*, which is a set of adjective pairs that contains at least one adjective pair in each cycle in the graph, using the method introduced by (Eades et al., 1993); note that this method divides the original directed graph into two DAGs, and we selected the larger DAG. The resulting DAG was then used to reformulate an adjective-ordering matrix, with the entry for an *unordered* adjective pair (s, t) carried over from the source adjective-ordering matrix if s and t were both present in the DAG. This process was repeated for each subset of adjective pairs, yielding four new subsets labeled *acyclic-noun*, *acyclic-verb*, *acyclic-all* and *acyclic-pure* (see Table 2 for details).

5 Experiment

We evaluated two different latent factor model-based Collaborative Filtering methods (Hofmann, 2004; Koren et al., 2009), *Singular Value Decomposition* (SVD) and *Non-negative Matrix Factorization* (NMF), on the task of predicting, given adjective pair data in the training set, whether a given adjective pair (in the test set) is ordered, and if it is ordered, which direction the ordering is in.[15]

Methodology. Given an adjective-ordering matrix, we trained each of the two CF models by employing nested 5-fold cross-validation with shuffling, in which the outer loop evaluates trained models, and the inner loop is used for model selection (hyperparameter tuning) and model fitting (i.e. training). Specifically, the outer-loop consists of 5-fold cross-validation, with 20% of the data (i.e. entries in the adjective-ordering matrix) held out as a test

[15]We used the implementations of NMF and SVD provided in the (python) library *Surprise* (v1.1.1) (Hug, 2020). See Appendix A for details about the computing infrastructure, software libraries and runtime used for these experiments.

dataset and the remaining data used for training and validation; the inner-loop consists of 5-fold cross-validation with 80% of the data used as a training set and the other 20% of the data held out as a validation set.[16] We evaluated performance during model selection by computing the mean average error (MAE), a metric that is commonly used for evaluating model-based CF algorithms.[17]

We trained and evaluated each of the two CF-models on each of the eight adjective-ordering matrices listed in Table 2. Note that a trained CF model, M, consists of: (i) a mapping, u_M, between (lemmatized) adjectives and embedding vectors of length n_f+1 – this mapping encodes the *first* item in an adjective pair (i.e. the s in an adjective pair (s,t)) into an embedding vector; (ii) a second mapping, v_M, between (lemmatized) adjectives and embedding vectors of length n_f+1 – this mapping encodes the *second* item in an adjective pair into an embedding vector; (iii) an $(n_f+1)\times(n_f+1)$ matrix, S_M – in the case of NMF, S_M is the identity matrix. The model estimates the value for an adjective bigram (from the test set), (s,t), as:

$$M(s,t) = u_M(s)S(v_M(t))^T$$

Given two adjectives a and b, the procedure for determining adjective ordering (which encapsulates the CF-model M) makes a prediction, $P_{\{a,b\}}$, about the ordering relationship between a and b as:

$$P_{\{a,b\}} = \begin{cases} A > B, & \text{if } M(a,b) \geq \psi > M(b,a) \\ A < B, & \text{if } M(a,b) < \psi \leq M(b,a) \\ \text{No Ordering}, & \text{Otherwise} \end{cases}$$

Here ψ is a threshold with $1 \leq \psi \leq 2$, such that we classify $M(s,t)$ as a *high* value (2) if $M(s,t) \geq \psi$ and a *low* value (1) otherwise. As the accuracy of the model depends on the value ψ, we thus evaluated model-performance by computing the Area Under the Receiver Operating Characteristic (AUROC) curve (Fawcett, 2006).

Results. Table 2 summarizes the results of our experiments. Notably, the CF models, NMF and SVD, achieved high AUROCs of 0.87 and 0.86 (respectively) on the *acyclic-all* adjective-pair data, and 0.79 and 0.78 (respectively) on the *all* adjective-pair data. Overall, NMF achieved the highest AUROC on each of the adjective pairs datasets. We also observed that both NMF and SVD performed better on the *acyclic* datasets than on their cyclic counterparts, and that for datasets both with and without cycles, both CF-models performed better on larger and less restricted datasets, *all* and *noun* (c.f. the smaller, more restricted datasets, *pure*).

We also evaluated a baseline model, referred to as the γ baseline, that serves as a reference point against which we compared the performance of the CF models. The γ baseline, which takes into account both input adjectives, is defined as follows. For an adjective z, let $\rho_{z,1}$ be the multiset of values for entries in the training data[18] where the first adjective is z, let $\rho_{z,2}$ be the multiset of values for entries in the training data where the second adjective is z, and let $h(z)$ be the weighted harmonic mean of $avg(\rho_{z,1})$ and $(3 - avg(\rho_{z,2}))$, so that:

$$h(z) = \frac{|\rho_{z,1}| + |\rho_{z,2}|}{|\rho_{z,1}|(\overline{\rho_{z,1}})^{-1} + |\rho_{z,2}|\,(3 - (\overline{\rho_{z,2}}))^{-1}}$$

Here, $h(z)$ is a ranking of the adjective z that is intended to correlate with z's position in the hierarchy of adjective-classes.[19] Given adjective pair (s,t) in the test data, the γ baseline is a linear transform[20] of the difference between the rankings of the two adjectives s and t:

$$\gamma(s,t) = \frac{3}{2} + \frac{1}{2}(h(s) - h(t))$$

Importantly, by using the γ baseline (in place of the CF models), our procedure can predict the ordering of adjectives, s and t, by comparing their rankings (per h) – importantly, this grounds the γ baseline in the second category of work described in §2.

Notably, the two CF models, NMF and SVD, outperformed the γ-baseline on each of the eight adjective-ordering matrices (see Table 2). We also observed that the γ baseline performed better on the *acyclic* datasets than on the full datasets – this was not surprising as the γ baseline is a model of a linear hierarchy of adjectives (i.e. $h(z)$ forms a *total preordering* over adjectives). Overall, our results show that both model-based CF algorithms: (i) perform well on the task of predicting adjective ordering, and (ii) outperform a baseline model

[16] An adjective-ordering matrix Q can be represented as a set of tuples of the form $(A, B, Q_{A,B})$ where A is an adjective coding for a row, $r_A \in Q$, B is an adjective coding for a column, $c_B \in Q$, and $Q_{A,B}$ is the value $Q[r_A, c_B]$ – then (A, B) is the model's input, and $Q_{A,B}$ is the model's output.

[17] Model selection for both models, NMF and SVD, involved optimizing the hyperparameter for the number of latent factors, $n_f \in \{4, 6, 8, \ldots, 16, 18\}$, and both models were trained for 450 epochs. The NMF model used a regularization rate of 0.06, and the SVD model used a learning rate of 0.005 and a regularization rate of 0.02.

[18] The training data does not include any zero-valued entries in the adjective-ordering matrix from which it is derived.

[19] N.b. the greater the number of adjectives that z precedes (as measured by $\overline{\rho_{z,1}}$), and the fewer the number of adjectives that precede z (as measured by $3-\overline{\rho_{z,2}}$), the larger $h(z)$ is.

[20] As $h(z) \in [1,2]$, this transform ensures $\gamma(s,t) \in [1,2]$.

grounded in a ranking (of adjectives) meant to correlate with a (linear) hierarchy of adjective-classes.

Analysis. We analyzed the degree to which the CF models, when predicting adjective ordering, utilize information about related adjective pairs. Let ω be the set of adjective pairs (in the training data), and let ψ be the set of all adjectives appearing in ω. Given an adjective pair (s, t), we define its *adjacent adjective pairs* (AAP) as the set $\mu_{(s,t)} \cap \omega$ where:

$$\mu_{(s,t)} = \{(x, y) \in \psi \times \psi | (s, y) \in \omega \wedge (x, t) \in \omega\}$$

We define the *fraction of adjacent adjective pairs* (FAAP) for (s, t) as the ratio:

$$|\mu_{(s,t)} \cap \omega| / |\mu_{(s,t)}|$$

Given an adjective pair (s, t), FAAP is the ratio of AAP present in the training data vs. the maximum number of AAP that could have been in the training data; the smaller the FAAP of (s, t) is, the more discriminating s and t are in the adjectives they appear with (in a bigram).

We now consider, for each (model, dataset) pair, how FAAP relates to model performance. We first computed the threshold[21] that maximized the model's F1-score, and then determined, for each adjective pair (s, t) in the test data, whether the model's output, $M(s, t)$, was correct (as classified by $P_{s,t}$).[22] We then computed the means and variances for the distributions of FAAP for adjective pairs that were correctly and incorrectly (respectively) classified, and we found that the mean of the former distribution was *consistently lower* than the mean of the latter.[23] We thus evaluated whether these two distributions of FAAP differed substantively. We used Welch's t-test (Welch, 1947) to test the null-hypothesis (α=0.01) that there is no (statistically) significant difference between the means of the two distributions (e.g. see Fig. 2); in each case, the null-hypothesis was rejected as the p-value was at most 3.5×10^{-9}, which was well below the critical value (α). We inferred that, consistently, the means of the two distributions differ significantly.

Moreoever, the t-test statistic appeared to correlate with the model's maximum F1-score, and is generally greater in the acyclic datasets (cf. the cyclic datasets) and in the *all* and *noun* subsets (cf. *verb* and *pure*). To validate this observation, we used linear regression to analyze the correlation between the t-test statistic of a model and the optimal F1-score of the model, which yielded a coefficient of determination (R^2) of 0.61 (p=3.7×10^{-4}). This suggests that the difference between the means of the correct and incorrect FAAP distributions is a significant factor in explaining model performance.

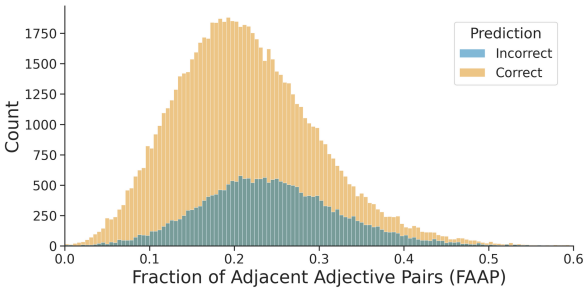

Figure 2: Given the ordering classifications made by the NMF model on the *all* test set, the distributions of FAAP for *correctly* and *incorrectly* classified adjective pairs have (mean, variance) of (0.237, 0.008) and (0.253, 0.008) respectively. Welch's t-test yields a test statistic of 28.97 and a p-value of 1.93×10^{-183}, which falls well below a critical p-value of 0.01 – hence, the means of the two distributions differ significantly.

6 Conclusion

This study showed that CF models, when trained on adjective bigram data readibly obtained from a corpus, perform well on the task of predicting the ordering of (unseen) adjective pairs. We compared the CF models with a baseline model, γ, that is grounded in a ranking of adjectives intended to parallel a (linear) hierarchy of adjective-classes. Notably, our results show that the CF models (consistently) perform markedly better than the baseline model, suggesting that CF models leverage adjective-ordering data that does not neatly align with proposed hierarchies of adjective-classes.

More broadly, the present study was motivated by the desire to see how far *productive* corpus-based models can be taken when they: (i) are restricted to adjective bigram statistics, (ii) do not require retention of the entire adjective-cooccurrence matrix after training, (iii) are robust in the face of sparse datasets, and (iv) must make predictions about the ordering of previously unseen adjective pairs. Moreoever, we made minimal assumptions about a learner's innate knowledge of language, the kind of data they have access to, and the size of their memory. To this extent, our procedure is a baseline that other models should aim to surpass to better justify any stronger assumptions they make.

[21] I.e. the threshold used to classify the model's (continuous) output as 1 (low) or 2 (high).

[22] We used, for each (model, dataset) pair, the model instance with median AUROC during cross-validation.

[23] Table 3 in the appendix details these distributions and lists the maximum F1-score for each (model, dataset) pair.

References

Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

Artemis Alexiadou. 2001. Adjective syntax and noun raising: Word order asymmetries in the dp as the result of adjective distribution. *Studia linguistica*, 55(3):217–248.

Mark Baker and William Croft. 2017. Lexical categories: Legacy, lacuna, and opportunity for functionalists and formalists. *Annual Review of Linguistics*, 3:179–197.

Mark C Baker and Mark Cleland Baker. 2003. *Lexical categories: Verbs, nouns and adjectives*, volume 102. Cambridge University Press.

Galia Bar-Sever, Rachael Lee, Gregory Scontras, and Lisa Pearl. 2018. Little lexical learners: Quantitatively assessing the development of adjective ordering preferences. In *Proceedings of the 42nd annual Boston university conference on language development*, pages 58–71. Cascadilla Press Somerville, MA.

Judy B Bernstein. 1993. *Topics in the syntax of nominal structure across Romance*. City University of New York.

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-based systems*, 46:109–132.

Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370.

Robin Burke. 2007. Hybrid web recommender systems. *The adaptive web*, pages 377–408.

Guglielmo Cinque. 1994. On the evidence for partial n-movement in the romance dp. In *Paths towards universal grammar. Studies in honor of Richard S. Kayne*. Georgetown University Press.

Guglielmo Cinque. 2010. *The syntax of adjectives: A comparative study*, volume 57. MIT press.

Guglielmo Cinque. 2014. The semantic classification of adjectives. a view from syntax. *Studies in Chinese Linguistics, 35.(1) 3-32*.

Guglielmo Cinque and Luigi Rizzi. 2012. The cartography of syntactic structures. In *The Oxford handbook of linguistic analysis*, pages 51–65. Oxford University Press.

Robert M. W. Dixon. 1982. Where have all the adjectives gone? *Berlin: Mouton de Gruyter*.

Peter Eades, Xuemin Lin, and William F Smyth. 1993. A fast and effective heuristic for the feedback arc set problem. *Information Processing Letters*, 47(6):319–323.

Ingrid Lossius Falkum. 2015. The how and why of polysemy: A pragmatic account. *Lingua*, 157:83–99.

Tom Fawcett. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

Didier L Goyvaerts. 1968. An introductory study on the ordering of a string of adjectives in present-day english. *Philologica Pragensia*, 11(1):12–28.

Michael Hahn, Judith Degen, Noah D Goodman, Dan Jurafsky, and Richard Futrell. 2018. An information-theoretic explanation of adjective ordering preferences. In *CogSci*.

Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53.

Felix Hill. 2012. Beauty before age? applying subjectivity to automatic english adjective ordering. In *Proceedings of the NAACL HLT 2012 student research workshop*, pages 11–16.

Thomas Hofmann. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 22(1):89–115.

Nicolas Hug. 2020. Surprise: A python library for recommender systems. *Journal of Open Source Software*, 5(52):2174.

Sagar Indurkhya. 2020. Inferring minimalist grammars with an SMT-solver. *Proceedings of the Society for Computation in Linguistics*, 3(1):476–479.

Sagar Indurkhya. 2021. Using collaborative filtering to model argument selection. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 629–639.

Sagar Indurkhya. 2022. Incremental acquisition of a Minimalist Grammar using an SMT-solver. In *Proceedings of the Society for Computation in Linguistics 2022*, pages 212–216, online. Association for Computational Linguistics.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Christopher Laenzlinger. 2005. French adjective ordering: Perspectives on dp-internal movement types. *Lingua*, 115(5):645–689.

Randy J LaPolla and Chenglong Huang. 2004. Adjectives in qiang. *Adjective classes: A cross-linguistic typology*, pages 306–322.

Jun Yen Leung, Guy Emerson, and Ryan Cotterell. 2020. Investigating Cross-Linguistic Adjective Ordering Tendencies with a Latent-Variable Model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4016–4028, Online. Association for Computational Linguistics.

Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. 2012. Syntactic annotations for the google books ngram corpus. In *Proceedings of the ACL 2012 system demonstrations*, pages 169–174.

Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. 2012. Recommender systems. *Physics Reports*, 519(1):1–49. Recommender Systems.

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Brian MacWhinney and Catherine Snow. 1985. The child language data exchange system. *Journal of child language*, 12(2):271–295.

Robert Malouf. 2000. The order of prenominal adjectives in natural language generation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 85–92.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Google Books Team, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, et al. 2011. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Luigi Rizzi and Guglielmo Cinque. 2016. Functional categories and syntactic theory. *Annual Review of Linguistics*, 2:139–163.

Alessandro Sanchez, Stephan C Meylan, Mika Braginsky, Kyle E MacDonald, Daniel Yurovsky, and Michael C Frank. 2019. childes-db: A flexible and reproducible interface to the child language data exchange system. *Behavior research methods*, 51(4):1928–1941.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.

J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer.

Gregory Scontras, Judith Degen, and Noah D Goodman. 2017. Subjectivity predicts adjective ordering preferences. *Open Mind*, 1(1):53–66.

Gregory Scontras, Judith Degen, and Noah D Goodman. 2019. On the grammatical source of adjective ordering preferences. *Semantics and Pragmatics*, 12:7.

Gary-John Scott. 2002. Stacked adjectival modification. *Functional Structure in DP and IP: The Cartography of Syntactic Structures, vol*, 1:91–122.

James Shaw and Vasileios Hatzivassiloglou. 1999. Ordering among premodifiers. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 135–143.

Ur Shlonsky. 2010. The cartographic enterprise in syntax. *Language and linguistics compass*, 4(6):417–429.

Richard Sproat and Chilin Shih. 1991. The cross-linguistic distribution of adjective ordering restrictions. In *Interdisciplinary approaches to language*, pages 565–593. Springer.

Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009.

Peter Svenonius. 1994. C-selection as feature-checking. *Studia linguistica*, 48(2):133–155.

Peter Svenonius. 2008. The position of adjectives and other phrasal modifiers in the decomposition of dp. In *Adjectives and Adverbs: Syntax, Semantics, and Discourse*.

John R Taylor. 2003. Polysemy's paradoxes. *Language sciences*, 25(6):637–655.

Nitzan Trainin and Einat Shetreet. 2021. It's a dotted blue big star: on adjective ordering in a post-nominal language. *Language, Cognition and Neuroscience*, 36(3):320–341.

Andreas Trotzke and Eva Wittenberg. 2019. Longstanding issues in adjective order and corpus evidence for a multifactorial approach. *Linguistics*, 57(2):273–282.

Zeno Vendler. 1968. Adjectives and nominalizations.

Bernard L Welch. 1947. The generalization of Student's 'problem when several different population varlances are involved. *Biometrika*, 34(1-2):28–35.

Charles Yang. 2011. Computational models of language acquisition. In *Handbook of generative approaches to language acquisition*, pages 119–154. Springer.

A Computing Infrastructure and Runtime

The experiments described in this paper were carried out on a linux server with the following specifications: Intel(R) Core(TM) i7-3930K CPU @ 3.20GHz; 54 GB of RAM; 1 TB of HDD. We used Python (v3.9.7) and the following libraries: *pandas* (v1.2.3) and *matplotlib* (v3.4.3), and *scipy* (v1.6.2); we used the implementation of Student's t-test and linear regression found in the python library, *scipy* (v1.6.2). The data-processing and experiment (i.e. model selection and training, evaluation and analysis-routines) took $\approx$30 hours of total runtime.

B Applications

The procedure introduced in this study makes minimal assumptions about a learner's innate knowledge of language, the kind of data they have access to, and the size of their memory. For this reason, we believe that upon further analysis of the procedure's performance on smaller corpora that reflect the primary linguistic data a child learner would encounter (MacWhinney and Snow, 1985; MacWhinney, 2000; Sanchez et al., 2019), our procedure may prove to be a suitable candidate for augmenting computational models of child language acquisition – see (Yang, 2011) for a review of computational models of language acquisition, and see (Indurkhya, 2020, 2022) for examples of models that this procedure could augment. Specifically, the procedure may be used to augment models that aim to explain how children acquire knowledge of restrictions on *constituent selection* (Svenonius, 1994) when forming syntactic structures, and in particular, when forming the *fine structure* of the Determiner Phrase (DP) in which a sequence of adjectives may be (hierarchically) embedded (Bernstein, 1993; Alexiadou, 2001; Baker and Croft, 2017). E.g. the productive model learned by this procedure can be used to score (or rank) candidate sequences of adjectival modifiers within a syntactic structure produced via a generative procedure, thereby serving as a filter on the production of adjective bigrams that the learner did not see in the primary linguistic data.

Model	Adj. Type	AUROC	Optimal F1-score	Optimal Threshold	FAAP for Correct Classification			FAAP for Incorrect Classification			t-test statistic	p-value
					Mean	Variance	Support	Mean	Variance	Support		
NMF	acyclic-all	0.867	0.794	1.345	0.215	0.007	78524	0.243	0.007	22978	45.505	0.000e+00
NMF	acyclic-pure	0.813	0.751	1.310	0.156	0.009	12433	0.167	0.010	4544	6.239	4.651e-10
NMF	acyclic-noun	0.858	0.809	1.365	0.280	0.007	21247	0.315	0.007	6171	29.530	7.752e-184
NMF	acyclic-verb	0.797	0.747	1.310	0.259	0.010	4156	0.289	0.009	1580	10.367	9.368e-25
NMF	all	0.788	0.764	1.375	0.237	0.008	84015	0.253	0.008	35554	28.972	1.926e-183
NMF	pure	0.756	0.736	1.280	0.165	0.009	12944	0.174	0.009	6355	5.915	3.414e-09
NMF	noun	0.767	0.773	1.370	0.297	0.007	22335	0.320	0.006	10026	23.431	8.223e-120
NMF	verb	0.716	0.726	1.210	0.285	0.011	4146	0.311	0.009	2514	10.426	3.208e-25
SVD	acyclic-all	0.862	0.794	1.395	0.214	0.007	78394	0.244	0.007	23113	46.278	0.000e+00
SVD	acyclic-noun	0.853	0.808	1.395	0.280	0.007	21132	0.312	0.007	6293	27.481	1.765e-160
SVD	acyclic-pure	0.809	0.748	1.390	0.153	0.008	12370	0.169	0.010	4567	9.344	1.201e-20
SVD	acyclic-verb	0.788	0.732	1.360	0.262	0.010	4047	0.289	0.009	1697	9.586	1.741e-21
SVD	all	0.770	0.757	1.370	0.235	0.008	81610	0.254	0.008	37983	33.420	4.391e-243
SVD	noun	0.734	0.766	1.300	0.296	0.007	21416	0.322	0.007	10924	26.799	9.675e-156
SVD	pure	0.738	0.727	1.270	0.163	0.009	12445	0.178	0.010	6806	10.143	4.372e-24
SVD	verb	0.689	0.724	1.155	0.278	0.010	4024	0.312	0.009	2644	13.654	8.414e-42

Table 3: A summary of the analysis of experiment results. Note that, for a given (model, dataset) pairing, the Optimal F1-score for the model was obtained by setting the model's classification threshold to the value listed under the Optimal Threshold. See §5 for additional details.