Semi-Supervised Modeling for Prenominal Modifier Ordering

Margaret Mitchell
University of Aberdeen
Aberdeen, Scotland, U.K.
m.mitchell@abdn.ac.uk

Aaron Dunlop
Oregon Health & Science University
Portland, OR
dunlopa@cslu.ogi.edu

Brian Roark
Oregon Health & Science University
Portland, OR
roark@cslu.ogi.edu

Abstract

In this paper, we argue that ordering prenominal modifiers – typically pursued as a supervised modeling task – is particularly well-suited to semi-supervised approaches. By relying on automatic parses to extract noun phrases, we can scale up the training data by orders of magnitude. This minimizes the predominant issue of data sparsity that has informed most previous approaches. We compare several recent approaches, and find improvements from additional training data across the board; however, none outperform a simple n-gram model.

1 Introduction

In any given noun phrase (NP), an arbitrary number of nominal modifiers may be used. The order of these modifiers affects how natural or fluent a phrase sounds. Determining a natural ordering is a key task in the surface realization stage of a natural language generation (NLG) system, where the adjectives and other modifiers chosen to identify a referent must be ordered before a final string is produced. For example, consider the alternation between the phrases "big red ball" and "red big ball". The phrase "big red ball" provides a basic ordering of the words *big* and *red*. The reverse ordering, in "red big ball", sounds strange, a phrase that would only occur in marked situations. There is no consensus on the exact qualities that affect a modifier's position, but it is clear that some modifier orderings sound more natural than others, even if all are strictly speaking grammatical.

Determining methods for ordering modifiers prenominally and investigating the factors underlying modifier ordering have been areas of considerable research, including work in natural language processing (Shaw and Hatzivassiloglou, 1999; Malouf, 2000; Mitchell, 2009; Dunlop et al., 2010), linguistics (Whorf, 1945; Vendler, 1968), and psychology (Martin, 1969; Danks and Glucksberg, 1971). A central issue in work on modifier ordering is how to order modifiers that are unobserved during system development. English has upwards of 200,000 words, with over 50,000 words in the vocabulary of an educated adult (Aitchison, 2003). Up to a quarter of these words may be adjectives, which poses a significant problem for any system that attempts to categorize English adjectives in ways that are useful for an ordering task. Extensive in-context observation of adjectives and other modifiers is required to adequately characterize their behavior.

Developers of automatic modifier ordering systems have thus spent considerable effort attempting to make reliable predictions despite sparse data, and have largely limited their systems to order modifier pairs instead of full modifier strings. Conventional wisdom has been that direct evidence methods such as simple n-gram modeling are insufficient for capturing such a complex and productive process.

Recent approaches have therefore utilized increasingly sophisticated data-driven approaches. Most recently, Dunlop et al. (2010) used both discriminative and generative methods for estimating class-based language models with multiple-sequence alignments (MSA). Training on manually curated syntactic corpora, they showed excellent in-domain performance relative to prior systems, and decent cross-domain generalization.

However, following a purely supervised training approach for this task is unduly limiting and leads to conventional assumptions that are not borne out in practice, such as the inapplicability of simple n-

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 236–241,
Portland, Oregon, June 19-24, 2011.

gram models. NP segmentation is one of the most reliable annotations that automatic parsers can now produce, and may be applied to essentially arbitrary amounts of unlabeled data. This yields orders-of-magnitude larger training sets, so that methods that are sensitive to sparse data and/or are domain specific can be trained on sufficient data.

In this paper, we compare an n-gram language model and a hidden Markov model (HMM) constructed using expectation maximization (EM) with several recent ordering approaches, and demonstrate superior performance of the n-gram model across different domains, particularly as the training data size is scaled up. This paper presents two important results: 1) N-gram modeling performs better than previously believed for this task, and in fact surpasses current class-based systems.[1] 2) Automatic parsers can effectively provide essentially unlimited training data for learning modifier ordering preferences. Our results point the way to larger scale data-driven approaches to this and related tasks.

2 Related Work

In one of the earliest automatic prenominal modifier ordering systems, Shaw and Hatzivassiloglou (1999) ordered pairs of modifiers, including adjectives, nouns ("baseball field"); gerunds, ("running man"); and participles ("heated debate"). They described a direct evidence method, a transitivity method, and a clustering method for ordering these different kinds of modifiers, with the transitivity technique returning the highest accuracy of 90.67% on a medical text. However, when testing across domains, their accuracy dropped to 56%, not much higher than random guessing.

Malouf (2000) continued this work, ordering prenominal adjective pairs in the BNC. He abandoned a bigram model, finding it achieved only 75.57% prediction accuracy, and instead pursued statistical and machine learning techniques that are more robust to data sparsity. Malouf achieved an accuracy of 91.85% by combining three systems. However, it is not clear whether the proposed ordering approaches extend to other kinds of modifiers, such as gerund verbs and nouns, and he did not present analysis of cross-domain generalization.

[1] But note that these approaches may still be useful, e.g., when the goal is to construct general modifier classes.

Dataset	2 mods	3 mods	4 mods
WSJ 02-21 auto	10,070	1,333	129
WSJ 02-21 manu	9,976	1,311	129
NYT	1,616,497	191,787	18,183

Table 1: Multi-modifier noun phrases in training data

Dataset	2 mods	3 mods	4 mods
WSJ 22-24	1,366	152	20
SWBD	1,376	143	19
Brown	1,428	101	9

Table 2: Multi-modifier noun phrases in testing data

Later, Mitchell (2009) focused on creating a class-based model for modifier ordering. Her system mapped each modifier to a class based on the frequency with which it occurs in different prenominal positions, and ordered unseen sequences based on these classes. Dunlop et al. (2010) used a Multiple Sequence Alignment (MSA) approach to order modifiers, achieving the highest accuracy to date across different domains. In contrast to earlier work, both systems order full modifier strings.

Below, we evaluate these most recent systems, scaling up the training data by several orders of magnitude. Our results indicate that an n-gram model outperforms previous systems, and generalizes quite well across different domains.

3 Corpora

Following Dunlop et al. (2010), we use the Wall St. Journal (WSJ), Switchboard (SWBD) and Brown corpus sections of the Penn Treebank (Marcus et al., 1993) as our supervised training and testing baselines. For semi-supervised training, we automatically parse sections 02-21 of the WSJ treebank using cross-validation methods, and scale up the amount of data used by parsing the New York Times (NYT) section of the Gigaword (Graff and Cieri, 2003) corpus using the Berkeley Parser (Petrov and Klein, 2007; Petrov, 2010).

Table 1 lists the NP length distributions for each training corpus. The WSJ training corpus yields just under 5,100 distinct modifier types (without normalizing for capitalization), while the NYT data yields 105,364. Note that the number of NPs extracted from the manual and automatic parses of the WSJ are quite close. We find that the overlap between the two groups is well over 90%, suggesting that extract-

ing NPs from a large, automatically parsed corpus will provide phrases comparable to manually annotated NPs.

We evaluate across a variety of domains, including (1) the WSJ sections 22-24, and sections commensurate in size of (2) the SWBD corpus and (3) the Brown corpus. Table 2 lists the NP length distributions for each test corpus.

4 Methods

In this section, we present two novel prenominal modifier ordering approaches: a 5-gram model and an EM-trained HMM. In both systems, modifiers that occur only once in the training data are given the Berkeley parser OOV class labels (Petrov, 2010).

In Section 5, we compare these approaches to the one-class system described in Mitchell (2010) and the discriminative MSA described in Dunlop et al. (2010). We refer the interested reader to those papers for the details of their learning algorithms.

4.1 N-Gram Modeling

We used the SRILM toolkit (Stolcke, 2002) to build unpruned 5-gram models using interpolated modified Kneser-Ney smoothing (Chen and Goodman, 1998). In the testing phase, each possible permutation is assigned a probability by the model, and the highest probability sequence is chosen.

We explored building n-gram models based on entire observed sequences (sentences) and on extracted multiple modifier NPs. As shown in Table 3, we found a very large (12% absolute) accuracy improvement in a model trained with just NP sequences. This is likely due to several factors, including the role of the begin string symbol <s>, which helps to capture word preferences for occurring first in a modifier sequence; also the behavior of modifiers when they occur in NPs may differ from how they behave in other contexts. Note that the full-sentence n-gram model performs similarly to Malouf's bigram model; although the results are not directly comparable, this may explain the common impression that n-gram modeling is not effective for modifier ordering. We find that syntactic annotations are critical for this task; all n-gram results presented in the rest of the paper are trained on extracted NPs.

Training data for n-gram model	Accuracy
Full sentences	75.9
Extracted multi-modifier NPs	88.1

Table 3: Modifier ordering accuracy on WSJ sections 22-24, trained on sections 2-21

4.2 Hidden Markov Model

Mitchell's single-class system and Dunlop et. al's MSA approach both group tokens into position clusters. The success of these systems suggests that a position-specific class-based HMM might perform well on this task. We use EM (Dempster et al., 1977) to learn the parameterizations of such an HMM.

The model is defined in terms of state transition probabilities $\mathrm{P}(c' \mid c)$, i.e., the probability of transitioning from a state labeled c to a state labeled c'; and state observation probabilities $\mathrm{P}(w \mid c)$, i.e., the probability of emitting word w from a particular class c. Since the classes are predicting an ordering, we include hard constraints on class transitions. Specifically, we forbid a transition from a class closer to the head noun to one farther away. More formally, if the subscript of a class indicates its distance from the head, then for any i, j, $\mathrm{P}(c_i \mid c_j) = 0$ if $i \geq j$; i.e., c_i is stipulated to never occur closer to the head than c_j.

We established 8 classes and an HMM Markov order of 1 (along with start and end states) based on performance on a held-out set (section 00 of the WSJ treebank). We initialize the model with a uniform distribution over allowed transition and emission probabilities, and use add-δ regularization in the M-step of EM at each iteration. We empirically determined δ smoothing values of 0.1 for emissions and 500 for transitions. Rather than training to full convergence of the corpus likelihood, we stop training when there is no improvement in ordering accuracy on the held-out dataset for five iterations, and output the best scoring model.

Because of the constraints on transition probabilities, straightforward application of EM leads to the transition probabilities strongly skewing the learning of emission probabilities. We thus followed a generalized EM procedure (Neal and Hinton, 1998), updating only emission probabilities until no more improvement is achieved, and then training both emission and transition probabilities. Often, we

Training data		WSJ Accuracy				SWBD Accuracy				Brown Accuracy			
		Ngr	1-cl	HMM	MSA	Ngr	1-cl	HMM	MSA	Ngr	1-cl	HMM	MSA
WSJ	manual	88.1	65.7	87.1	87.1	72.9	44.7	71.3	71.8	67.1	31.9	69.2	71.5
	auto	87.8	64.6	86.7	87.2	72.5	41.6	71.5	71.9	67.4	31.3	69.4	70.6
NYT	10%	90.3	75.3	87.4	88.2	84.2	71.1	81.8	83.2	81.7	62.1	79.5	80.4
	20%	91.8	77.2	87.9	89.3	85.2	72.2	80.9	83.1	82.2	65.9	78.9	82.1
	50%	92.3	78.9	89.7	90.7	86.3	73.5	82.2	83.9	83.1	67.8	80.2	81.6
	all	92.4	80.2	89.3	92.1	86.4	74.5	81.4	83.4	82.3	69.3	79.3	82.0
NYT+WSJ auto		93.7	81.1	89.7	92.2	86.3	74.5	81.3	83.4	82.3	69.3	79.3	81.8

Table 4: Results on WSJ sections 22-24, Switchboard test set, and Brown test set for n-gram model (Ngr), Mitchell's single-class system (1-cl), HMM and MSA systems, under various training conditions.

find no improvement with the inclusion of transition probabilities, and they are left uniform. In this case, test ordering is determined by the class label alone.

5 Empirical results

Several measures have been used to evaluate the accuracy of a system's modifier ordering, including both type/token accuracy, pairwise accuracy, and full string accuracy. We evaluate full string ordering accuracy over all tokens in the evaluation set. For every NP, if the model's highest-scoring ordering is identical to the actual observed order, it is correct; otherwise, it is incorrect. We report the percentage of orders correctly predicted.

We evaluate under a variety of training conditions, on WSJ sections 22-24, as well as the testing sections from the Switchboard and Brown corpus portions of the Penn Treebank. We perform no domain-specific tuning, so the results on the Switchboard and Brown corpora demonstrate cross-domain applicability of the approaches.

5.1 Manual parses versus automatic parses

We begin by comparing the NPs extracted from manual parses to those extracted from automatic parses. We parsed Wall Street Journal sections 02 through 21 using cross-validation to ensure that the parses are as errorful as when sentences have never been observed by training.

Table 4 compares models trained on these two training corpora, as evaluated on the manually-annotated test set. No system's accuracy degrades greatly when using automatic parses, indicating that we can likely derive useful training data by automatically parsing a large, unlabeled training corpus.

5.2 Semi-supervised models

We now evaluate performance of the models on the scaled up training data. Using the Berkeley parser, we parsed 169 million words of NYT text from the English Gigaword corpus (Graff and Cieri, 2003), extracted the multiple modifier NPs, and trained our various models on this data. Rows 3-6 of Table 4 show the accuracy on WSJ sections 22-24 after training on 10%, 20%, 50% and 100% of this data. Note that this represents approximately 150 times the amount of training data as the original treebank training data. Even with just 10% of this data (a 15-fold increase in the training data), we see across the board improvements. Using all of the NYT data results in approximately 5% absolute performance increase for the n-gram and MSA models, yielding roughly commensurate performance, over 92% accuracy. Although we do not have space to present the results in this paper, we found further improvements (over 1% absolute, statistically significant) by combining the four models, indicating a continued benefit of the other models, even if none of them best the n-gram individually.

Based on these results, this task is clearly amenable to semi-supervised learning approaches. All systems show large accuracy improvements. Further, contrary to conventional wisdom, n-gram models are very competitive with recent high-accuracy frameworks. Additionally, n-gram models appear to be domain sensitive, as evidenced by the last row of Table 4, which presents results when the 1.8 million NPs in the NYT corpus are augmented with just 11 thousand NPs from the WSJ (auto) collection. The n-gram model still outperforms the other systems, but improves by well over a percent, while the class-based HMM and MSA approaches

are relatively static. (The single-class system shows some domain sensitivity, improving nearly a point.)

5.3 Cross-domain evaluation

With respect to cross-domain applicability, we see that, as with the WSJ evaluation, the MSA and n-gram approaches are roughly commensurate on the Brown corpus; but the n-gram model shows a greater advantage on the Switchboard test set when trained on the NYT data. Perhaps this is due to higher reliance on conventionalized collocations in the spoken language of Switchboard. Finally, it is clear that the addition of the WSJ data to the NYT data yields improvements only for the specific newswire domain — none of the results change much for these two new domains when the WSJ data is included (last row of the table).

We note that the improvements observed when scaling the training corpus with in-domain data persist when applied to very diverse domains. Interestingly, n-gram models, which may have been considered unlikely to generalize well to other domains, maintain their superior performance in each trial.

6 Discussion

In this paper, we demonstrated the efficacy of scaling up training data for prenominal modifier ordering using automatic parses. We presented two novel systems for ordering prenominal modifiers, and demonstrated that with sufficient data, a simple n-gram model outperforms position-specific models, such as an EM-trained HMM and the MSA approach of Dunlop et al. (2010). The accuracy achieved by the n-gram model is particularly interesting, since such models have previously been considered ineffective for this task. This does not obviate the need for a class based model — modifier classes may inform linguistic research, and system combination still yields large improvements — but points to new data-rich methods for learning such models.

Acknowledgments

This research was supported in part by NSF Grant #IIS-0811745 and DARPA grant #HR0011-09-1-0041. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the NSF or DARPA.

References

Jean Aitchison. 2003. *Words in the mind: an introduction to the mental lexicon*. Blackwell Publishing, Cornwall, United Kindgom, third edition. p. 7.

Stanley Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report, TR-10-98, Harvard University.

Joseph H. Danks and Sam Glucksberg. 1971. Psychological scaling of adjective order. *Journal of Verbal Learning and Verbal Behavior*, 10(1):63–67.

Arthur Dempster, Nan Laird, and Donald Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38.

Aaron Dunlop, Margaret Mitchell, and Brian Roark. 2010. Prenominal modier ordering via multiple sequence alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL (HLT-NAACL 2010)*, pages 600–608, Los Angeles, CA, USA. Association for Computational Linguistics.

David Graff and Christopher Cieri. 2003. *English Gigaword*. Linguistic Data Consortium, Philadelphia, PA, USA.

Robert Malouf. 2000. The order of prenominal adjectives in natural language generation. In *Proceedings of the 38th ACL (ACL 2000)*, pages 85–92, Hong Kong.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

J. E. Martin. 1969. Semantic determinants of preferred adjective order. *Journal of Verbal Learning and Verbal Behavior*, 8(6):697–704.

Margaret Mitchell. 2009. Class-based ordering of prenominal modifiers. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 50–57, Athens, Greece. Association for Computational Linguistics.

Margaret Mitchell. 2010. A flexible approach to class-based ordering of prenominal modifiers. In E. Krahmer and M. Theune, editors, *Empirical Methods in Natural Language Generation*, volume 5980 of *Lecture Notes in Computer Science*. Springer, Berlin / Heidelberg.

Radford M. Neal and Geoffrey E. Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Michael I. Jordan, editor, *Learning in Graphical Models*. Kluwer Academic Publishers, Dordrecht.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American*

Chapter of the ACL (HLT-NAACL 2007), pages 404–411, Rochester, NY, USA. Association for Computational Linguistics.

Slav Petrov. 2010. Berkeley parser. GNU General Public License v.2.

James Shaw and Vasileios Hatzivassiloglou. 1999. Ordering among premodifiers. In *Proceedings of the 37th ACL (ACL 1999)*, pages 135–143, College Park, Maryland. Association for Computational Linguistics.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP 2002)*, volume 2, pages 901–904.

Zeno Vendler. 1968. *Adjectives and Nominalizations*. Mouton, The Netherlands.

Benjamin Lee Whorf. 1945. Grammatical categories. *Language*, 21(1):1–11.