A Deep Context Grammatical Model For Authorship Attribution

Simon Fuller, Phil Maguire, Philippe Moser
CS Dept, National University of Ireland, Maynooth
Maynooth, Kildare, Ireland
SIMON.FULLER.2010@nuim.ie, pmaguire@cs.nuim.ie, pmoser@cs.nuim.ie

Abstract

We define a variable-order Markov model, representing a Probabilistic Context Free Grammar, built from the sentence-level, de-lexicalized parse of source texts generated by a standard lexicalized parser, which we apply to the authorship attribution task. First, we motivate this model in the context of previous research on syntactic features in the area, outlining some of the general strengths and limitations of the overall approach. Next we describe the procedure for building syntactic models for each author based on training cases. We then outline the attribution process – assigning authorship to the model which yields the highest probability for the given test case. We demonstrate the efficacy for authorship attribution over different Markov orders and compare it against syntactic features trained by a linear kernel SVM. We find that the model performs somewhat less successfully than the SVM over similar features. In the conclusion, we outline how we plan to employ the model for syntactic evaluation of literary texts.

Keywords: Authorship Attribution; Syntactic Features; Markov Models

1. Introduction

Syntactic features for authorship attribution have received considerable attention recently. Parts of speech tags have been studied extensively, (see Stamatatos (2009) and Luyckx (2010) for comprehensive overviews), Gamon (2004) trained an SVM over a transition rule feature set, Luyckx and Daelemans (2008) have used shallow parsing, Kaster et al. (2005) have examined the role of parse depth in classification, Feng et al. (2012) have established the efficacy of employing a number of different deep grammatical features in an SVM classifier, and van Cranenburgh (2012) successfully employs a tree-kernel SVM. Raghavan et al. (2010) train a Probabilistic Context Free Grammar (PCFG) for each author, and then parse test cases using these models, choosing the model which parses with highest probability.

Purely syntactic approaches have generally been found inferior to traditional lexical approaches in terms of pure attribution accuracy. However, there is some evidence (Feng et al., 2012; Raghavan et al., 2010) that combining syntactic features with traditional lexical features is superior to either approach when used alone.

Stamatatos (2009) describes a number of other shortcomings with the syntactic approach, including the language dependence of the parsing/tagging tools required in order to prepare the source text, and the introduction of error/noise by the parsing process. The latter is of particular concern for applications to social media contexts where loose grammar, slang, abbreviations and acronyms are common (and hence where character ngrams are perhaps most resilient).

Regarding Raghavan's particular approach, Feng et al. (2012) observe that since lexical leaf production rules are constituents of the PCFG model employed, it is difficult to assess the relative discriminative powers of the lexical and syntactic feature components. As such, it is not clear from their study the extent to which syntactic features *per se* contribute to classification. Also, if off-the-shelf PCFGs are used, then each text needs to be reparsed for each author model, which is computationally expensive.

One particular motivation for removing lexical information is that syntactic features are among those most relevant to traditional literary analysis. Therefore it is potentially beneficial, in the context of computational studies of literary texts, to employ models which discriminate via features similar to those employed in general comparative literature studies. Hence syntactic features have been of particular interest to researchers who have focused on literature (Feng et al., 2012; Jautze et al., 2013).

2. Deep Context Grammatical Modeling

We take an approach which is closest to Raghavan et al. (2010). We retain the core premise of creating author-specific PCFGs and assigning texts to whichever model parses with the highest probability. We abstract from lexical information simply by building our own model and discarding the lexical level.

We develop this approach by exploiting conditional probabilities through *vertical Markovization*, i.e. increasing the Markov order to consider ancestors preceding the production rule, effectively counting at increasing depths the context wherein a rewrite rule occurs. This technique has been employed extensively in PCFG parsers, for instance by Johnson (1998), Collins (2003), Charniak (2000), and Klein and Manning (2003), and 2-order Markov feature sets have been used for authorship attribution by Feng et al. (2012). Intuitively, these techniques are relevant to authorship attribution of literary texts, since sentence structure features prominently in traditional literary analysis, and previous studies (Feng et al., 2012; Jautze et al., 2013) have demonstrated that sentence structural forms differ significantly between cases in author and genre studies.

We build a sentence-deep variable-order Markov model similar to a generalized suffix tree, which we call a *generalized parse-suffix tree*, and compare different probability estimators at varying Markov orders.

Our purpose in this article is to define the model and demonstrate its efficacy as a representation of an individual author's style. To do so, we compare the attribution

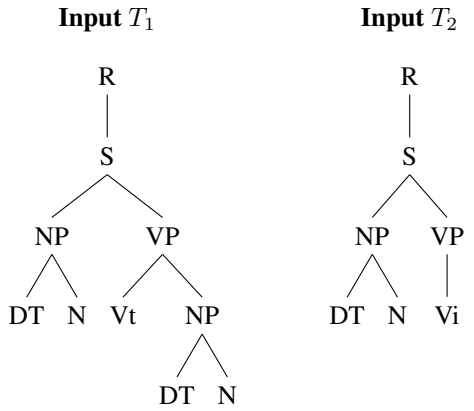

Figure 1: Trees T_1 and T_2: example inputs to the model

accuracy to that of a linear kernel SVM over broadly equivalent syntactic features, which is the basic model employed by Feng et al. (2012). While our model performs slightly below the latter, the proximity of the accuracy of the model to this benchmark suggests that a uniform-weighted high-order Markov model captures most of the syntactic variance between authors, and is thus a strong and intuitive formal model for further stylometric literary modeling and analysis.

3. Model And Algorithms

The data set's training and test cases are consecutive sequences of sentence-level parse trees with the lexical levels removed.[1] We train a model for each author with grammars drawn from a set of their respective works. The total number of tags per text is set to a fixed number, by curtailing the depth of the final input tree. We then estimate the probability of a test case given each model, and return the author whose model maximizes the probability.

3.1. Model

Each input tree starts at a root node with reserved label R, from which descends the estimated grammatical parse of the sentence with the lexical leaves removed. We record the counts of every production rule under all Markov orders to sentence depth. Upon insertion of each input tree, we increment the model by these counts, recording the occurrence of rules in contexts of all available depths. **Table 1** provides the table and **Figure 2** the tree structure, after insertion of T_1 and T_2.

For efficient insertion and retrieval, we distribute these counts across a suffix-tree augmented with production counts at internal nodes. A tree is inserted as follows: for each node in the tree (uniquely identified by its label and ancestors), insert it into the model at the equivalent location to which it appears in the tree, and then recursively insert it into the equivalent nodes under progressively lower Markov orders, until the node is added to the root, R. Inserting a node has two steps: i) if the node's label (e.g. NP) does not exist as a model node in the required location already, then create a node with that label and locally record its depth; ii) in the dictionary attached to that model node, increment by one the value associated with the key representing that node's children, i.e. the production rule activated at that node in the input tree.

The final detail is the use of suffix pointers to traverse upwards to equivalent nodes at lower Markov orders. These are used to perform the recursive insertions quickly, and later to switch efficiently during probability estimation. **Figure 2** illustrates the model after T_1 and T_2 are inserted. For clarity, only suffix pointers for tag DT are shown.

[1] We employ a standard parser – Stanford PCFG Lexicalized, English model(Klein and Manning, 2003).

Table 1					
L	**Rule**	**C**	**L**	**Rule**	**C**
R:1	S	2	VP:3	Vt, NP	1
R:1	DT, N	3	S:4	NP, VP	2
R:1	NP, VP	2	NP:5	DT, N	1
R:1	Vi	1	NP:6	DT, N	2
R:1	Vt, NP	1	VP:7	Vt, NP	1
NP:2	DT, N	3	VP:7	Vt	1
VP:3	Vi	1	NP:8	DT, N	1

Table 1: **L**, the location in **Figure 2** where the count is stored; **Rule**: the specific rewrite; and **C**, the counts, upon insertion of trees T_1 and T_2. For instance the rewrite rule NP $\rightarrow$ DT, N occurs 3 times at NP(2) and twice at NP(6) in **Figure 2**.

3.2. Probability Estimation

The task is to assign a given text S to an author-model M_i. Let M_i, $1 \leq i \leq k$ be k author models. S contains n sentences, s_j which are calculated sequentially, $\mathbf{P}\ (S|M_i)$ being the product of the probability of its sentences under M_i, i.e. $\mathbf{P}\ (S|M_i) = \prod_1^n \mathbf{P}\ (s_j|M_i)$.

Using this, we apply Bayes' rule, i.e.:

$$\mathbf{P}(M_i|S) = \mathbf{P}(S|M_i) * \mathbf{P}(M_i)/\mathbf{P}(S).$$

We drop the denominator and attribute the text to the highest scoring model M_t, where:

$$t = \underset{\text{i}}{\operatorname{argmax}}\, \mathbf{P}(S|M_i) * \mathbf{P}(M_i).$$

Since in the test cases we here consider, every author has an equal number of cases in the training and test sets, we drop the prior and this simplifies here to:

$$t = \underset{\text{i}}{\operatorname{argmax}}\, \mathbf{P}(S|M_i).$$

Generally, priors representing a known disproportion, or adaptive priors such as Dirichlet, can be employed without altering $\mathbf{P}(S|M_i)$. We now describe $\mathbf{P}(S|M_i)$.

The probability of a sequence is the product of its sentences, and the probability of a parsed sentence is the product of the probability of its rules. The probability of a given rule is estimated at the position in the model-tree that corresponds to its position in the input tree, when this depth is less than or equal to the Markov assumption. When the rule's occurrence in the input tree is deeper than the Markov assumption, we traverse the suffix-pointers (**Figure 2**) until we reach the desired depth. Now we return the count for the

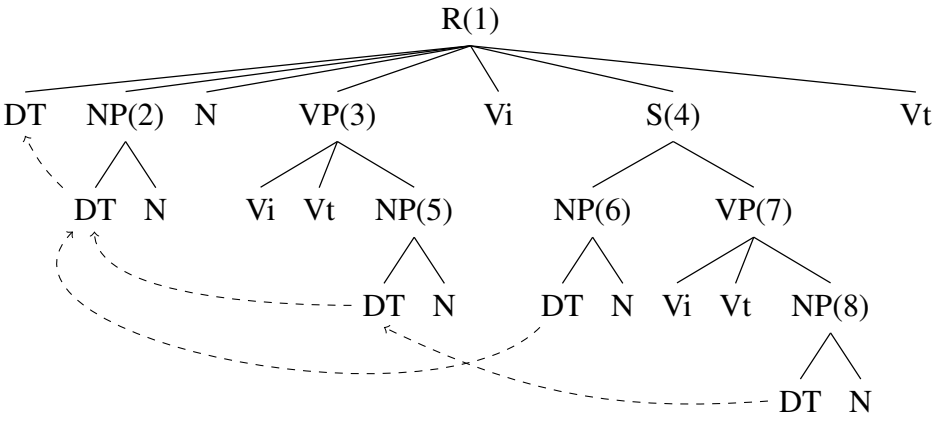

Figure 2: Parse suffix tree after insertion of T_1 and T_2 - only the Vi nodes are added with T_2. The internal nodes are numbered corresponding to the **L** columns in **Table 1**, e.g. the count **C** (⟨ DT, N ⟩ | VP, NP) is stored at NP(5). For clarity we include only the suffix-pointers for DT nodes, represented by the dashed arrows.

rule at the node over the total count at that node (see **Table 1**). We here employ 5 probability estimations Q_1, Q_2, Q_3, Q_4 and Q_5, which assume Markov orders of 1 to 5 successively (or the highest possible lower order given the node's depth in the input tree), where a production rule is order 1.

As a final point, since a node may occur in a test sentence which is not represented in the model, we use a zero-frequency estimator to give a small probability for an unseen occurrence. Preliminary investigations explored back-off estimators (Katz, 1987) and PPM escape estimators (Cleary and Witten, 1984; Moffat, 1990). However it proved more effective to penalize cases of non-occurrences of a context more heavily than these estimators permit. Hence we opt for the heuristic that a non-occurring tag's probability is its model-wide frequency over the total number of parsed tags, and we keep positive occurrences as described above.

4. Tests: Authorship Attribution

Following standard practice, we examine performance on two separate sample sets: the first consists of 10 works by 10 19th Century/early 20th Century novelists drawn from Project Gutenberg (www.gutenberg.org).[2]; the second consists of 10 works by contemporary suspense/mystery writers.[3]

We parsed the texts using the Stanford NLP English model and discarded lexical levels. We randomly separated them into 5 groups, of 2 works per author, and performed 4 vs 1 cross-validation across these 5 groups, of which we report the mean accuracy. Employing the 5 Q_i probability models described above, we sampled text from the start of each document, rising in increments of 1000 tags to 5000.

As we state in the introduction, lexical approaches have been consistently found to outperform syntactic ones, and we do not re-evaluate these against our own model, since the general trend has been well demonstrated in the previ-

[2]Conrad, Dickens, Hardy, Scott, Eliot, Southworth, James, Gaskell, Stevenson, Trollope.

[3]Baldacci, Coben, Grisham, Koontz, Patterson, Cook, Cornwell, Crichton, Archer, Follett.

Table 2A: Classics - PCFG					
Sample Size	Q_1	Q_2	Q_3	Q_4	Q_5
1000	49	47	51	48	45
2000	53	49	59	59	47
3000	71	70	76	76	71
4000	78	82	86	84	77
5000	80	84	86	87	85
Table 2B: Classics - SVM					
Sample Size	Q_1	Q_2	Q_3	Q_4	Q_5
1000	39	39	44	36	37
2000	50	53	50	58	48
3000	65	59	70	74	67
4000	76	78	85	84	74
5000	84	85	89	86	78

Table 2: Mean % accuracies of 4 vs 1 cross-validation on sample set of 10 classic novelists for PCFG model and Linear SVM.

Table 3A: Suspense Writers - PCFG					
Sample Size	Q_1	Q_2	Q_3	Q_4	Q_5
1000	46	44	50	49	52
2000	58	63	63	59	58
3000	67	67	66	68	70
4000	71	71	69	75	79
5000	78	73	76	84	87
Table 3B: Suspense Writers - SVM					
Sample Size	Q_1	Q_2	Q_3	Q_4	Q_5
1000	58	55	46	44	37
2000	72	72	64	55	50
3000	74	78	80	71	60
4000	83	82	80	77	68
5000	89	88	85	85	69

Table 3: Mean % accuracies of 4 vs 1 cross-validation on sample set of 10 contemporary suspense novelists for PCFG model and Linear SVM.

ous literature. Rather, we compare our model against an SVM over similar feature sets, in order to evaluate the relative performance of the algorithms themselves, rather than the relative performance of different classes of features, i.e. lexical and syntactic. Stamatatos (2009) cites SVMs as among the best machine learning algorithms for authorship attribution, and a linear SVM has been previously been employed by Feng et al. (2012) over syntactic features, so we chose this as our benchmark for evaluating the algorithm.
Specifically, we used LIBLINEAR through R (Fan et al., 2008; Helleputte, 2010) with the Crammer and Singer (2001) form, re-selecting the 3000 most common features across all training sets for each cross-validation. We normalized to document frequency over chosen features, then scaled according to the training set. We selected features from the model according to Markov orders 1 to 5, where an order 1 feature is a production rule, and higher orders include ancestors. **Table 2** presents the respective results for the Classics set and **Table 3** for the contemporary fiction.

5. Results

Our model performs slightly below the SVM at highest sample sizes, and significantly below at some smaller sizes. This is somewhat expected since the SVM weights different features in order to differentiate the candidate training sets. The SVM responds to higher orders inconsistently, and performance deteriorates for the highest, probably due to the small sample sizes for these more complex features. The PCFG model generally improves with Markov order at the larger sample sizes, i.e. given enough sample data. For instance in **Table 3** the PCFG reaches 87% at order 5 from 78% at order 1, while the SVM peaks at 89% at order 1. Hence, given a more advanced learning algorithm such as an SVM, deeper syntactic features have comparatively limited effect. Our model tacitly imposes uniform weighting across the tree structure, implicitly claiming that a simple and consistent grammatical relation can differentiate authors. For this simpler model, with stronger constraints, the increased Markov order generally improves performance, indicating that the deep sentence structure of authors is a determinable characteristic, and that authors become more distinguishable when their models are compared at greater depths.
We have only carried out a perfunctory process of feature selection for the SVM, and a more rigorous and comprehensive process of selection and pruning would most likely produce better results. Nonetheless our relatively simple model produces competitive results, whilst making stronger theoretical claims regarding author differentiation, e.g. in its uniform weight attribution as described above.

6. Conclusion

We have described a variable-order PCFG model and demonstrated its general efficacy as a syntactic classifier by comparing its performance to a linear SVM. This high-order Markov model over the deep syntactic structure of a collection of classes of texts successfully models most of the syntactic difference between the classes. For future work we will make use of this model for stylometric analysis of authors and genres, for instance to return characteristic sentence and phrase structures for different classes, drawn from the suffix-tree models, which will be compared to traditional stylistic comparisons of the authors under examination. We also plan to create a variable weighted version of the PCFG to increase attribution accuracy.

7. Acknowledgements

This research is partly funded by an Irish Research Council Embark scholarship awarded to Simon Fuller. We also thank the Computer Science Department of NUI Maynooth for helping to fund this work. We further thank the LREC reviewers for their comments and suggestions.

8. References

Charniak, E. (2000). A maximum-entropy-inspired parser In Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference (NAACL), pp. 132–139.

Chen, Y & Lin, C. (2006). Combining SVMs with Various Feature Selection Strategies. In Feature Extraction, Foundations and Applications Ed. Guyon et. al. SprinSger, Berlin Heidelberg New York

Cleary, J.G. & Witten, I.H. (1984). Data compression using adaptive coding and partial string matching. In IEEE Transactions on Communications 32(4): 396–402.

Collins, M. (2003). Head-Driven Statistical Models for Natural Language Parsing. In Computational Linguistics 29(4): 589–637.

Crammer, K. & Singer, Y. (2001). On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. In Journal of Machine Learning Research 2: (2001) 265-292

Cranenburgh, A. van (2012). Literary authorship attribution with phrase-structure fragments In Proceedings of the 2012 Computational Linguistics for Literature workshop, Montreal, Canada, June 8. pp. 59-63, Montreal, Canada

Fan, R. & Chang, K. & Hsieh, C. & Wang, X. & Lin, C. (2008). LIBLINEAR: A Library for Large Linear Classification, In Journal of Machine Learning Research 9: 1871–1874

Feng, S. & Banerjee, R. & Choi, Y. (2012). Characterizing Stylistic Elements in Syntactic Structure. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 1522-1533, Jeju Island, Korea.

Gamon, M. (2004). Linguistic correlates of style: authorship classification with deep linguistic analysis features. In Proceedings of the 20th International Conference on Computational Linguistics (COLING), pp. 611–617, Geneva Switzerland.

Gusfield, D. (1997). Algorithms on Strings, Trees and Sequences. Cambridge University Press, Cambridge, UK.

Helleputte, T. (2010). LiblineaR: Linear Predictive Models Based On The Liblinear C/C++ Library, R package http://www.thibaulthelleputte.be/

Jautze, K. & Koolen, C. & Cranenburgh, A. van & de Jong, H. (2013). From high heels to weed attics: a syntactic

investigation of chick lit and literature In Proceedings of the Workshop on Computational Linguistics for Literature pp. 72–81, Atlanta, Georgia.

Johnson, M. (1998). PCFG Models of Linguistic Tree Representations. In Computational Linguistics 24(4): 613–632.

Kaster, A. & Siersdorfer, S. & Weikum, G. (2005). Combining Text and Linguistic Document Representations for Authorhip Attribution. In SIGIR Workshop: Stylistic Analysis of Text for Information Access (STYLE).

Katz, S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. In IEEE Transactions on Acoustics, Speech, and Signal Processing, 35(3): 400–401

Klein D. & Manning, C. (2003). Accurate Unlexicalized Parsing. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), pp. 423–430.

Luyckx, K. (2010). Scalability Issues in Authorship Attribution. University Press Antwerp.

Luyckx K. & Daelemans, W. (2008). Using Syntactic Features to Predict Author Personality from Text. In: Proceedings of Digital Humanities 2008, pp. 146–149, Oulu, Finland

Moffat, A. (1990). Implementing the PPM Data Compression Scheme. In IEEE Transactions on Communications 38(11): 1917–1921.

Raghavan, S. & Kovashka, A. & Mooney, R. (2010). Authorship Attribution Using Probabilistic Context-Free Grammars. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 38–42.

Stamatatos, E. (2009). A Survey of Modern Authorship Attribution Methods In Journal of the American Society for Information Science and Technology 60(3): 538–556.