Identifying Distinct Components of a Multi-Author Document

Navot Akiva
Dept. of Computer Science
Bar Ilan University
Ramat Gan, Israel
navot.akiva@gmail.com

Moshe Koppel
Dept. of Computer Science
Bar Ilan University
Ramat Gan, Israel
moishk@gmail.com

Abstract— **Given a multi-author document, we use unsupervised methods to identify distinct authorial threads. Although this problem is of great practical interest for security and forensic reasons, as well as for commercial purposes, this paper is, to the best of our knowledge, the first presentation of a general-purpose method for solving it.**

Keywords-Authorship Attribution, Document Clustering, Text Mining

I. INTRODUCTION

Suppose we are faced with a text of interest for security reasons, which is suspected to be an amalgam of sources. For such potential multi-source document we wish to determine which parts of the document originate from each source, in order to analyze each source separately with regard to content and provenance. Such determination might also be of commercial or legal interest, as in the case of contemporary documents, or of academic or cultural interest, as in the case of important historical documents.

Remarkably, there has been very little work on this problem reported in the literature. The single paper that we know of that attacked this exact problem [1] is limited to a very special class of documents, such as the Bible, for which considerable linguistic resources, including manual translations and concordances, are available.

In this paper, we offer a new algorithm that solves the authorship decomposition problem for all types of documents. (Note that for our purpose here, we regard each source as an author, although in principle the division into distinct sources might reflect other distinctions.) We do not assume that any other writing of the authors is available to us. The general idea of our method is to chunk the document, vectorize the chunks and cluster the vectors. Then we use the clusters as the basis for supervised learning and apply the learned models to each sentence of the document. We will demonstrate the effectiveness of our method even where the document covers multiple themes that are distributed identically across the writings of the respective authors. Furthermore, unlike the work of [1], we consider also the case where there may be more than two authors.

The structure of the paper is as follows. In the next section, we sketch some related work. In Section 3, we specify our problem in a bit more detail, explain the experimental setup and describe the corpora on which we test our method. In Section 4 we explain our method and in Section 5 we offer results.

II. PREVIOUS WORK

Several active areas of research are closely related to the problem we attack in this paper. Clustering a set of anonymous documents according to authorship has been explored recently in [2]. However, this work differs from ours in that its objective is to cluster multiple documents, each of which is written by a single author. By contrast, we wish to segment a single unsegmented document according to authorship, with no guarantees that any unit longer than a single sentence is written by a single author.

A number of papers [3][4][5][6] have considered the problem of segmenting documents according to topic. This is a considerably easier task than the one we consider here since the differences between segments dealing with different topics is more salient than the differences between segments written by different authors about the same topic.

The work of [7] considered the problem of decomposing a document according to authorship, but their method was a supervised one: they had labeled examples of same-author and different-author pairs of paragraphs by the two authors. We have no such labeled examples; indeed our method is entirely unsupervised.

Some research has considered the problem of clustering topics and authors using Latent Dirichlet Analysis [8]. However, that work assumes that each author's work reflects a distinct distribution over topics. In this paper, we explicitly reject that assumption. In fact, in at least one of our testbeds, the respective distributions over topics for the authors are identical.

In addition, some work on intrinsic plagiarism detection [9][10] also attempts to segment a document according to authorship, but it assumes an asymmetric split: there is a single dominant author and the object is to identify outlier segments. We make no such assumption.

Finally, as mentioned, [1] offered a solution to our problem that depended on manual resources that could be used to identify and disambiguate occurrences of synonyms in a text. Our approach eliminates the need for such resources and goes beyond their approach in several other ways as well, as will be seen below.

III. Experimental Setup

A. Problem Definition

Formally, the problem we wish to solve is this: we have a single document written by k co-authors. We assume that any given sentence was written by exactly one of the authors and that generally the authors took turns writing reasonably long sequences of sentences. The number of authors, k, is known to us. Our objective is to segment the text and to cluster the resulting segments in accordance with the parts of the text written by the respective co-authors .

B. Test Corpus Generation Method

To test our methods, we artificially create a multi-author text using the following procedure. Begin with k documents (or corpora), $T_1,\ldots,T_k$, each by a distinct single author. Informally, at each stage we choose a random number of sentences from the work of a randomly selected author, $T_1,\ldots,T_k$, and append it to the merged text we are creating.

More formally, for i=1,2,…

1) Choose document $T_{t(i)}$, where t(i) is a randomly chosen integer from 1 to k subject to the condition that $t(i)\neq t(i-1)$. This is the author from whom we will now draw the next sequence of sentences.
2) Append to the new text the first x unused sentences in $T_{t(i)}$, where x is randomly chosen from a uniform distribution over the integers from 1 to n. Note that n is the maximal length of a sequence of sentences drawn from a single author. It determines the granularity of the merged text.

The procedure continues until all k documents have been exhausted.

Note that, because we use a uniform distribution, the lengths of the single-author segments (that is, the number of sentences between transitions) might vary very widely. This makes the problem more difficult – and more realistic.

C. Evaluation Corpora

We will apply this method to three different corpora.

Our first corpus consists of pairs of biblical books. We choose this corpus both for its intrinsic cultural and historical interest and for purposes of comparing our method to that of [1], which was tested only on this corpus.

It might be argued that clustering merged bible texts leverages possible topical coherence of the individual authorial components and hence does not prove the effectiveness of our methods for cases where authorial components are thematically indistinguishable. Hence, our second corpus consists of blog posts by the economist Gary Becker and the law scholar Richard Posner. Becker and Posner maintain a joint blog[1] in which they each post weekly on a mutually agreed upon topic. We choose a random number of sentences from Becker, followed by a random number of sentences from Posner, and so on. (We pay no attention to boundaries between individual posts by any given author.) Segmenting the resulting merged text into Becker segments and Posner segments presents two challenges. First, the wide variety of topics offers many plausible ways to split the text along thematic lines. Second, the aggregate texts by Becker and by Posner, respectively, are *identically distributed over topics*, so that no thematic clues can help us to distinguish between the respective authors.

Finally, we use a corpus consisting of four columnists writing for the New York Times (Gail Collins, Thomas Friedman, Maureen Dowd and Paul Krugman). This corpus offers the same challenges as in the case of the Becker-Posner blog, but in addition allows us to tackle the case where k>2.

IV. Our Method

A. Algorithm

We first present the method in broad outline, leaving the details for the exposition below. Given a merged text and the desired number of authorial threads, k, do the following:

1) Chunk the text into segments of some fixed length l.
2) Represent each such chunk as a vector that encodes some essential lexical features of that chunk. The determination of these features is crucial, as we'll see below.
3) Measure the similarity of every pair of chunks (or, more specifically, their vector representations).
4) Cluster the chunks into k clusters using some clustering algorithm.
5) Regard each chunk as being labeled by its cluster assignment and use the labeled chunks as training examples for learning a classifier that assigns a text to one of the k classes.
6) Use the learned classifier to tentatively assign each individual sentence in the merged text to one of the k classes.
7) Every sentence that is assigned to some class with high confidence (as will be defined) is locked into that class. Each other sentence is assigned in accordance with the nearest confidently assigned sentences before and after it.

The central idea is that we represent chunks of the document in terms of features likely to capture authorship and cluster accordingly. Then these clusters can be used as the basis for supervised learning (in conjunction with a wider range of features than used in the initial clustering) which yields a classifier that can be applied to individual sentences – thus resulting in the sort of fine-grained division that we seek.

[1] www.becker-posner-blog.com

B. *Becker-Posner as Example*

For clarity and specificity, let's run through the method as applied to the Becker-Posner blog.

Creating the merged document

Becker's text consists of 14,689 sentences and Posner's text consists of 12,233 sentences. We create a merged text by alternately taking sentences from each author, the number of sentences drawn from a uniform distribution between 1 and 200. In the resulting merged text, there are 261 transitions from one author to the other, with a median of just above 100 sentences between transitions (as we'd expect).

Chunking and vectorization

We begin to decompose the text into two clusters of sentences by chunking the text into segments of 40 sentences each. (Results aren't very sensitive to chunk size, as long as chunks are smaller than the median single-author run.) Thus, in this case we obtain 674 chunks. Of these chunks, 265 are pure Becker, 194 are pure Posner and the rest are mixed.

We now represent the chunks as vectors. Our feature set consists of the 500 most common words in the document. Each chunk is represented as a binary vector indicating whether the corresponding word does or does not appear in the chunk. The fact that we use only the most common words in the document is crucial. These common words are mostly function words or common content words that appear across many topics. If we were to use in our representation words that are tied to specific topics, the clustering is likely to distinguish topics rather than authors. Furthermore, the binary nature of the vector is also crucial. [1] reported that clustering failed completely when chunks were represented in terms of frequencies of common words; we have confirmed that result. However, we find that when *binary* representations are used, more salience is given to words that are almost never used by one author or the other and this is the key to obtaining clustering along lines of authorship. (For example, Becker almost never uses the word *thus*.)

Clustering

Next we use cosine to measure the similarity between each pair of chunks and use n-cut clustering [11] to cluster the chunks into two clusters. We find that of the 265 pure Becker clusters 263 are assigned to Cluster 1 and of the 194 pure Posner chunks 192 are assigned to Cluster 2.

Supervised Learning

Now comes the key step in which the clustered chunks serve as the basis for supervised learning. Using all 674 chunks as labeled training examples, we represent each example in terms of frequencies of all words that appear in the corpus at least five times and use SVM (linear; default settings) to learn a classifier to distinguish Class 1 text from Class 2 text. Note that the feature set used in this stage is far larger than the feature set used in the initial clustering. In the initial clustering it was important to use a small set of features that would be more likely to capture distinct writing styles than to capture distinct topics. However, now that we have established what we hope is a reasonable (though imperfect) clustering, we are satisfied to capture all distinctions between the clusters. For example, if it happens that different authors do have somewhat different topic preferences, that information can be exploited at this stage.

Classifying individual sentences

We then apply this classifier to all 26,922 sentences in the merged text. We find that 84% of Becker sentences are assigned to Class 1 and 87% of Posner sentences are assigned to Class 2. Now, we consider only the 25% of sentences that are assigned to one class or the other with highest confidence (reflected in distance from the SVM boundary). Of these 6730 sentences, 97% of Becker sentences are assigned to Class 1 and 98% of Posner sentences are assigned to Class 2. Finally, for any unassigned sentence S, if the last assigned sentence before S and the first assigned sentence after S are assigned to the same class, we assign S to that class. When there is a string of unassigned sentences following a sentence assigned to Class 1 and succeeded by a sentence assigned to Class 2, we identify the optimal split point and assign all sentences prior to the split point to Class 1 and all the rest to Class 2. (Optionally, borderline sentences can be left unassigned.)

Assigning all sentences, in our final split, 94% of Becker sentences are assigned to Class 1 and 95% of Posner sentences are assigned to Class 2.

V. Results

We apply the method exactly as just described to each of our testbed corpora. Our measure of success is as follows: we create a k*k confusion matrix M in which m_{ij} is the number of sentences of author i assigned to class j. After fixing the order of the authors, we order the classes in such manner as to maximize the sum of the diagonal (that is, we associate classes with authors in the most "natural" way). We define purity as the proportion of sentences on the diagonal [12] and use that as our measure of success.

A. *Biblical Pairs*

For our first experiment, we apply our method to four pairs of biblical books. In each case, two books from the same biblical sub-genre were merged and needed to be separated out. In Fig. 1, we show results for each pair using our method (right bar) and that of [1] (left bar). (Since they left some borderline sentences unassigned, we do the same, for comparison purposes.) The advantage that we claim for our method is its applicability to other genres; as for the biblical genre, we are satisfied that, as can be seen, our method performs almost as well as the method of [1] which was defined for that specific genre.

B. *Becker-Posner*

For our next experiment, we again consider the Becker-Posner blog. As we saw in the exposition above, the purity of our split for this problem is about 94% for the case where the median number of sentences between transitions is about 100. It might be wondered, however, how sensitive our results are to the granularity of the merged document. The problem presumably gets harder if transitions from one author to the other are more frequent. To test this, we vary the maximum number of

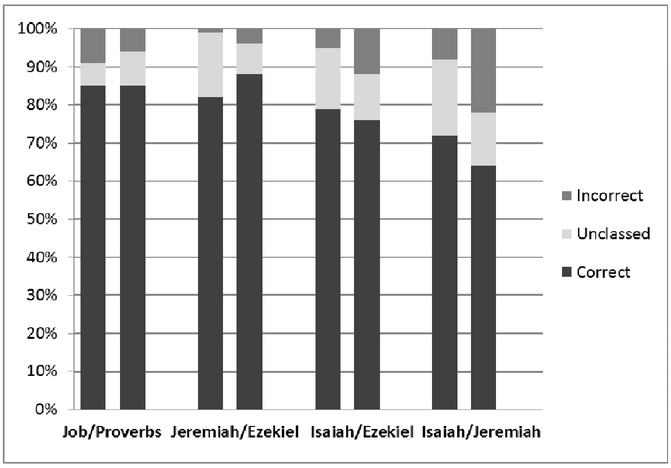

Figure 1. % of correctly assigned, unassigned and incorrectly assigned sentences for merged pairs of biblical books. For each pair, results for our method are on the right and those of [1] are on the left.

sentences between transitions from one author to another (the parameter n in Step 2 of our construction in section 3 above). In Fig. 2, we show results on Becker-Posner for various values of n (median distance between transitions is approximately n/2). Indeed we find that for higher granularity (that is, for lower values of n), purity diminishes somewhat.

C. *NYT Columnists*

Finally, we apply our method to the New York Times columnists corpus. Our purpose here is to apply our method to cases in which we have 3 or 4 authors participating in a single merged document. We note that for documents created by merging the writing of any of the six pairs of columnists, our results are in line with those we obtained for author pairs in our earlier experiments (purity ranging from 88% to 96%, depending on the pair). Results for documents created by merging more than two authors are shown in Fig. 3. We find that for documents with three authors, we obtain purity ranging from 77% to 82% and for a document with four authors, we obtain purity of 74%.

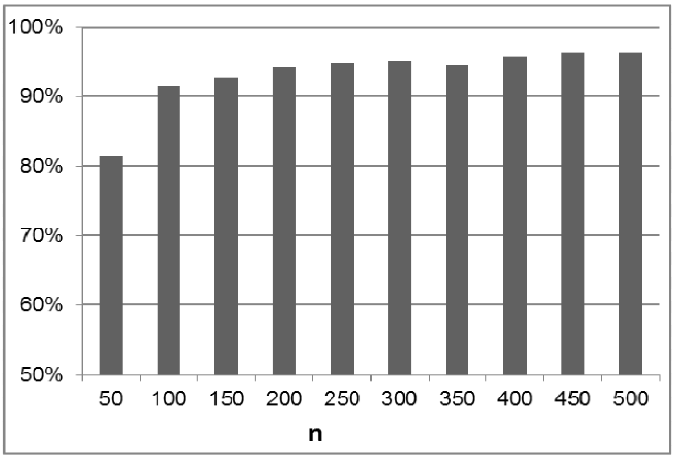

Figure 2. Purity of our obtained splits for merged Becker-Posner documents with various degrees of granularity.

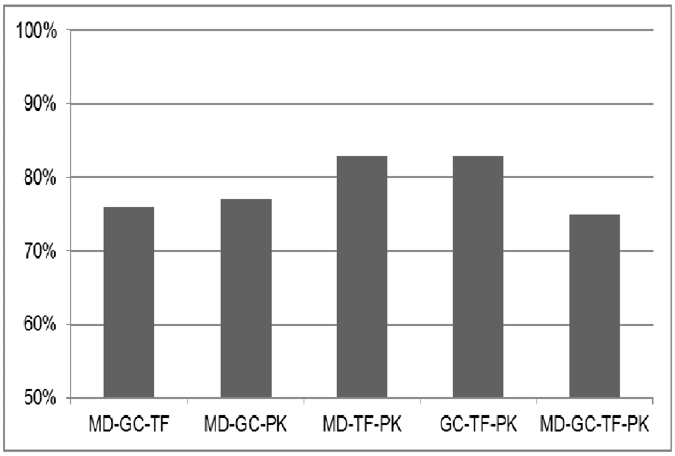

Figure 3. Purity of our obtained splits for documents created by merging three or four NY Times columnists (MD=Maureen Dowd, GC=Gail Collins, TF=Thomas Friedman, PK=Paul Krugman)

VI. Conclusions

We have presented what we believe is the first general-purpose method for decomposing an unsegmented multi-author document into its distinct authorial threads. Unlike previous work, we require no training data and no specialized corpus-specific linguistic resources. In fact, our method is language-independent.

We find that we can separate out authorial threads even when the distribution of topics across authors is identical (as in the case of Becker and Posner). Moreover, we obtain reasonable results even for decomposing documents with three or four authors.

Such decomposition could serve as a crucial pre-processing step in the analysis of suspicious documents for both content and provenance.

It remains a weakness of our approach that we assume that the number of participating authors is known in advance. The automatic determination of this number remains an open problem.

References

[1] M. Koppel, N. Akiva, I. Dershowitz and N. Dershowitz, "Unsupervised decomposition of a document into authorial components", proceedings of ACL, Portland OR, pp. 1356-1364. August 2011.

[2] R. Layton, P. Watters and R. Dazeley, "Automated unsupervised authorship analysis using evidence accumulation clustering", Natural Language Engineering, Available on CJO 2011 doi:10.1017/S1351324911000313.

[3] J. P. Callan, "Passage-level evidence in document retrieval", proceedings of the 17th Annual International ACM/SIGIR Conference, Dublin, Ireland, pages 302–310. 1994.

[4] F. Choi, "Advances in domain independent linear text segmentation", proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics, Seattle, WA, pages 26–33. 2000.

[5] M. H. Hearst, "Multi-paragraph segmentation of expository text", proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, NM, , pages 9–16. 1994.

[6] J. C. Reynar, "An automatic method of finding topic boundaries", proceedings of the Student Session of the 32nd Annual Meeting of the

Association for Computational Linguistics,Las Cruces, NM, pages 331–333. 1994.

[7] N. Graham, G. Hirst, and B. Marthi, "Segmenting documents by stylistic character", Natural Language Engineering, 11(4):397-415.2005.

[8] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smyth and M. Steyvers , “Learning author-topic models from text corpora”, ACM Transactions on Information Systems (TOIS), ACM, 2010.

[9] S. Meyer zu Eissen and B. Stein, "Intrinsic plagiarism detection", in Proc. of the 28th ECIR, pp. 565-569. London, 2006.

[10] E. Stamatatos, "Intrinsic plagiarism detection using character n-gram profiles", proc. of SEPLN’09 the 3rd Int. Workshop on Un-covering Plagiarism, Authorship, and Social Software Misuse. pp. 38–46. 2009.

[11] I. S. Dhillon, Y. Guan and B. Kulis, "Weighted graph cuts without eigenvectors: a multilevel approach", IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 29(11):1944-1957. 2007.

[12] C. D. Manning, P. Raghavan and H. Schütze, "Introduction to information retrieval", Cambridge University Press, 2008.