Identifying Earmarks in Congressional Bills

Ellery Wulczyn*
Stanford University
ellery@cs.stanford.edu

Madian Khabsa*
Microsoft Research
madian.khabsa@microsoft.com

Vrushank Vora
University of Chicago
vvora@uchicago.edu

Matthew Heston
Northwestern University
heston@u.northwestern.edu

Joe Walsh, Christopher Berry,
Rayid Ghani
University of Chicago
{jtwalsh,cberry,rayid}@uchicago.edu

ABSTRACT

Earmarks are legislative provisions that direct federal funds to specific projects, circumventing the competitive grant-making process of federal agencies. Identifying and cataloging earmarks is a tedious, time-consuming process carried out by experts from public interest groups. In this paper, we present a machine learning system for automatically extracting earmarks from congressional bills and reports. We first describe a table-parsing algorithm for extracting budget allocations from appropriations tables in congressional bills. We then use machine learning classifiers to identify budget allocations as earmarked objects with an out of sample ROC AUC score of 0.89. Using this system, we construct the first publicly available database of earmarks dating back to 1995. Our machine learning approach adds transparency, accuracy and speed to the congressional appropriations process.

1. INTRODUCTION

An earmark is a legislative provision that directs federal funds to specific projects, circumventing the competitive grant-making process of federal agencies. It has been difficult to study how earmarking has affected the legislative process due to a lack of comprehensive and open data on earmarks. In fact, earmarking is often considered "the best known, most notorious, and most misunderstood aspect of the congressional budgetary process" [14].

In the past, earmark datasets were created manually by experts from governmental agencies, such as the Office of Management and Budget (OMB) and public interest groups such as Taxpayers for Common Sense, Washington Watch, and Citizens Against Government Waste. Congress produces thousands of pages of legal text each year, making the process time-consuming and expensive. As a result, past efforts were limited to short time spans and a limited set of documents. A further issue, is that each of these groups had different definitions, motivations, and processes for identifying and cataloging earmarks.

Given the significant changes in American politics in the last 20 years, e.g. increased polarization, changes to campaign financing, and a recent ban on earmarks [3, 6], datasets with short-term coverage are inadequate for political scientists to draw robust policy conclusions. A consistent, historically complete dataset has the potential to reveal valuable insights on effective governance and answer questions such as: "How instrumental is earmarking to passing controversial legislation?"; "What effect does securing earmarks have on campaign financing and reelection?"; or "Does being a chair of congressional subcommittee affect how funds are appropriated to the legislator‘s state or district?"

In this paper, we present a machine learning system for automatically extracting earmarks from congressional bills and reports. This approach allows us to cheaply, reliably, and consistently extract earmarks from historical congressional documents. Furthermore, the method is transparent and reproducible, enabling analysts to easily understand, evaluate, and build on our work [7]. Using our system, we construct the first publicly available database of earmarks dating back to 1995.

The dataset is already being used by Harris School of Public Policy researchers of the University of Chicago to do public policy research and analysis. In addition, it is being promoted by both the Center for Data Science and Public Policy, also of the University of Chicago, as well as the Sunlight Foundation to researchers and practitioners interested in government transparency and public policy.

2. SYSTEM OVERVIEW

An overview of the entire system is depicted in Figure 1. The ultimate goal is to be able to take a document, extract all the potential budget allocations that occur in the document, and finally classify which allocations are earmarks. Extracting potential budget allocations involves identifying tables in the document that contain budget allocations and parsing them into rows, where each row is a separate allocation. This process is described in Section 4. Building a classifier for determining which allocations are earmarks requires a labeled set of allocations. We generate a corpus of labeled allocations by matching the extracted allocations to an earmark dataset compiled by the Office of Management and Budget (OMB) in the late 2000s (see Section 5.1). Our method for labeling earmarks involves a combination of hand-labeling and machine learning and is discussed in

[*] The first two authors contributed equally to this study.

KDD '16, August 13-17, 2016, San Francisco, CA, USA
© 2016 ACM. ISBN 978-1-4503-4232-2/16/08...$15.00
DOI: http://dx.doi.org/10.1145/2939672.2939711

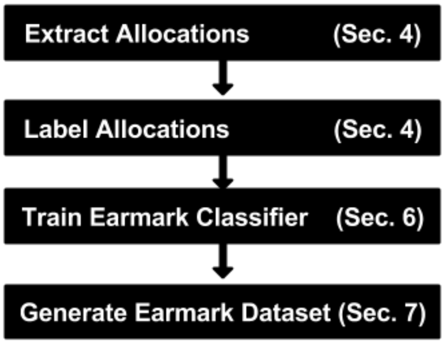

Extract Allocations (Sec. 4)
Label Allocations (Sec. 4)
Train Earmark Classifier (Sec. 6)
Generate Earmark Dataset (Sec. 7)

Figure 1: System overview.

Section 5. Given this labeled set of annotations, we train an earmark detection classifier as described in Section 6. Finally, we run our earmark classifier on all congressional bills and reports going back to 1995 and show some preliminary analysis in Section 7.

3. THE BUDGET PROCESS

As many readers may not be familiar with the congressional budgetary process and how earmarking occurs, we provide a brief overview.

At the beginning of each fiscal year, the President submits a proposal for the year's budget to Congress. The budget proposes funding levels for the various government entities and broadly outlines spending limits and revenue expectations for at least the next five years. Next, the House and Senate budget committees review the President's proposal and pass a budget resolution.

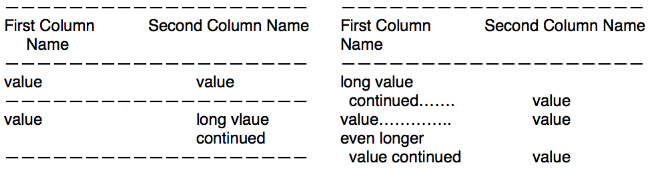
Mandatory spending and interest payments account for the majority of federal spending [2]. The House and Senate appropriations committees divide what remains in the budget resolution to their twelve sub-committees. The twelve sub-committees then write the bills that authorize discretionary spending, and each of those bills becomes a law if approved by each chamber and signed by the President. Congress combines those bills into a single omnibus bill and votes on it.

Earmarks can enter at almost any point of the process. Senators and representatives can insert earmarks into the text of appropriations bills, including supplemental appropriations and continuing resolutions [15]. they can place earmarks in the explanatory report attached to the bill. They can also contact bureaucrats directly.

As the budget process has changed over time, so has the placement of earmarks: "During the 19th century, earmarks were often placed in the law. But after the adoption of the Budget and Accounting Act of 1921, most earmarks were included in legislative reports" [8]. Congress officially banned earmarks in 2010, but members have continued to request and receive them [9, 4]. Senators and representatives have increasingly turned to calling and writing federal agencies directly [10].

4. ALLOCATION EXTRACTION

The first step to identifying earmarks is to extract appropriations found in congressional bills and reports. We focus on extracting allocations mentioned within tables, where 85% of the earmarks occur (see Sec. 5.4). Future work will identify earmarks in free text to increase the coverage of our dataset.



```
--------------------------------
First Column      Second Column Name
  Name
--------------------------------
value             value
--------------------------------
value             long vlaue
                  continued
--------------------------------
```

```
--------------------------------
First Column      Second Column Name
Name
--------------------------------
long value
  continued.......          value
value.............          value
even longer
  value continued           value
```

Figure 2: Examples of *dashed* and *dotted* tables.

Several approaches to table parsing have been developed in the field of information retrieval. Pyreddy et al. exploit table layout in text documents and develop a character alignment graph (CAG) that uses heuristic methods to identify tables within documents [13]. They identify sections of tables within documents. Pinto et al. extend the CAG to extract individual cells from tables [11]. Our initial analyses found that many tables shared similar attributes. We thus employ a heuristic based approach described in the following section.

4.1 Table Identification

The Government Printing Office (GPO) provides congressional bills and reports as plain text files where tables appear as blocks of formatted text. Indentation, white space, and dots and dashes are used to format tables. We first segment a document into paragraphs, where paragraphs are separated by two new line characters or more. Then each paragraph is classified as a table or free text. A paragraph is labeled as a table if the percentage of rows satisfying any of the following conditions exceeds a threshold:

- It has numeric characters and three consecutive dots,
- It has numeric characters and at least two consecutive spaces, or
- It has at least three consecutive dashes.

The threshold was set empirically to 0.3. In the experiments section (5.6), we show that this heuristic retrieved more than 98% of the tables in congressional reports and bills.

Furthermore, tables in Congressional bills and reports can be categorized into two main types: *dotted tables* and *dashed tables*. Dashed tables use lines of all dashes to separate rows and whitespaces to separate columns. Dotted tables do not have special lines separating rows: each line is a row or part of a multi-line row, and columns are separated by dots and whitespace. See Figure 2 for examples. We distinguish these two classes of tables because parsing each type requires different rules and heuristics.

4.2 Table Header Detection

Parsing a one-line table header requires splitting on two or more white spaces. In the case where headers span multiple lines, we introduce an algorithm that clusters words in the header based on their vertical overlap. The simple idea is that two words on two consecutive lines that intersect vertically belong to the same header. First, each line is split into cells by two or more consecutive white spaces.

Each cell in every row is represented as a four-dimensional tuple $(text, line, begin, end)$, where *text* is the clean text of the cell, *line* is the line number, *begin* is the offset within the line at which the text begins, and *end* is the last index within the line at which the cell ends. The tuples are fed into a clustering algorithm 1 to detect headers. The algorithm returns the list of headers of the table.

Algorithm 1: Header Identification Algorithm

```
input  : List of cell tuples $Cells$
output: Table Headers
$Sorted = Sort(Cells, key = begin)$;
$Clusters \longleftarrow List$;
$C \longleftarrow List$;
Add $Sorted[0]$ to $C$;
Add $C$ to $Clusters$;
for $i = 1; i < length(Sorted); i++$ do
    $word \longleftarrow Sorted[i]$;
    $prev\_word \longleftarrow Sorted[i-1]$;
    if $word.begin < prev\_word.end$ then
        Add $word$ to $Clusters[-1]$;
    else
        $C \longleftarrow List$;
        Add $word$ to $C$;
        Add $C$ to $Clusters$;
    end
end
$Headers \longleftarrow List$;
for $C \in Clusters$ do
    $header = Sort(C, key = line)$;
    Add $concat(header)$ to $Headers$;
end
return $Headers$;
```

4.3 Column Detection

Tables are treated as a collection of columns where each column divides rows into multiple cells. The idea behind detecting column dividers is that column boundaries do not contain any text. Instead, they consist of whitespace or other delimiting characters across all the rows. That is, a column divider is a pair of *begin* and *end* positions such that only whitespace is observed within this span for all the lines in the table. Algorithm 2 shows how these tuples are found. In the following section, we discuss how multiline rows are detected and merged.

4.4 Multiline Row Merging

Each line is initially treated as a separate row in the table. Because rows can span multiple lines, we develop heuristics to detect multiline rows and merge the related lines. For dashes tables, identifying multiline rows is trivial because rows are separated by a line of dashes. For dotted tables, things are more involved. The basic concept is that all valid table rows need to have a money allocation in one of their associated lines. This is guaranteed by our table-identification heuristic described earlier. In the case of multiline rows, money allocations can appear either in the first or the last line. Fortunately, the position of the allocation tends to be consistent within a table, which makes it easy to group multiline rows. This heuristic is shown to provide accurate extraction in the experiments section.

Algorithm 2: Column Identification Algorithm

```
input  : List of table rows $Rows$
$p \longleftarrow get\_white\_space\_positions(Rows[0])$;
for $row \in Rows$ do
    $p \longleftarrow p \cap get\_white\_space\_positions(row)$;
end
$indices \longleftarrow Sort(p)$;
$dividers \longleftarrow list$;
if $length(indices) = 0$ then
    Add $(0, length(Rows[0]))$ to $dividers$;
else
    $i \longleftarrow 1$;
    $begin \longleftarrow indices[0]$;
    $prev \longleftarrow indices[0]$;
    while $i < length(indices)$ do
        if $indices[i] - prev = 1$ then
            $prev \longleftarrow indices[i]$;
        else
            Add $(begin, prev)$ to $dividers$;
            $begin \longleftarrow indices[i]$;
            $prev \longleftarrow indices[i]$;
        end
        $i \longleftarrow i + 1$;
    end
end
return $dividers$;
```

4.5 Table Parsing Evaluation

To evaluate the the accuracy of our table-identification methodology, we randomly selected 40 documents and tagged all 384 tables in those documents. On this subset, our table-identification heuristic recalled 98.6% of all of true tables, while 89.2% of the predicted tables were true tables. For the table identification task, we value recall higher than precision because an earmark missed in this step will never be recovered and rows that are not allocations can be weeded out later. To evaluate the column-identification and row-merging algorithms, we randomly chose 30 tables from those 40 documents. We correctly identified columns and rows in all 30.

5. ALLOCATION LABELING

The output of the table parsing algorithm, described in Sec. 4, is a collection of tables that are neatly parsed into rows and columns. Each non-header row in an allocation table describes an allocation and is a potential earmark. Eventually, we want to take a supervised learning approach to building a model to classify allocations as earmarks. This approach, however, requires labeling a set of allocations as earmarks or just regular appropriations.

In this section, we describe how we use a corpus of earmarks from the Office of Management and Budget (OMB), described in Sec. 5.1, to generate a labeled set of allocations. In particular, we attempt to match each earmark in the OMB dataset to a corresponding allocation. In order to perform this matching task at scale, we first match a subset by hand (Sec. 5.4) and then train a classifier to predict whether a budget allocation matches an earmark in the OMB dataset (Sec. 5.5 and Sec. 5.6).

5.1 OMB Data

In 2007, the OMB ordered all departments and agencies to identify and catalog earmarks that appeared in appropriations and authorization bills and reports for 2005. Over the course of three months, those departments and agencies sent the OMB congressional funding data. The OMB used this to compile a list of congressional earmarks [12]. This process was repeated in 2008, 2009, and 2010. The OMB posted all the data in CSV format on its website.[1]

For each earmark, the OMB may provide the following:

- The congressional documents it appeared in,
- An excerpt from each of those documents,
- A short description,
- A long description, and
- The recipient of the funds.

See Table 1 for an example OMB record.

Documents	Congress 111, House Bill 3293
Excerpt	...of which $13,455,000 shall be used for the projects, and in the amounts, specified under the heading 'Disease Control, Research, and Training' in the report of the Committee on Appropriations of the House of Representatives to accompany this Act
Short Desc.	Dillard, University, New Orleans, LA for facilities and equipment
Full Desc.	NA
Recipient	NA

Table 1: Sample OMB Record.

5.2 Resolving OMB document references

The first step in matching OMB records to allocations is to map each document cited in the OMB corpus to a congressional document[2]. Unfortunately, there is no unique government document ID, which would make linking trivial. Instead, the OMB references the documents in which an earmark appears in myriad ways, such as specifying a bill number, a report number accompanying a bill, a public law, or a common name for a bill. Table 2 lists typical examples of references. Table 3 describes how references are resolved for OMB data from 2008. The heuristics for the other years are similar.

Earmark ID	**Citation Reference**
340045	H. Rept. 110-434
235530	P.L. 110-161
235531	Joint Explanatory Statement to accompany H.R. 2764

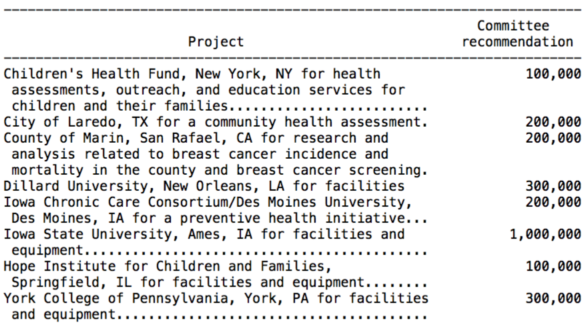
Table 2: Examples of OMB document references.

Citation Reference	**Earmark Document**
H.Rept. XXX-YYY; S.Rept. XXX-YYY	Mapped to the latest House Report in the XXX Congress that goes to the YYY report; mapping is similar for the Senate
H.R. XXX; S.XXX	Mapped to the latest version of the House or Senate bills of the XXX number in a given Congress and all of the accompanying reports to those bills
P.L. XXX-YYY	Public Laws are bills signed into law by the president of the U.S. We find the latest house version of the law and map it to the bills and all of the relevant documents.
Joint Explanatory Statements	These documents are treated as congressional reports to the bill they are attached to. We map them to those reports accordingly.

Table 3: OMB reference resolution for 2008.

5.3 Matching OMB Records to Allocations

After linking OMB document references to documents in our corpus of congressional texts, we used fields from an OMB record to link OMB records with allocations extracted from the corpus. For example, consider the OMB record in Table 1 and a table extracted from the cited document in Figure 3.



```
------------------------------------------------------------------------
                                                           Committee
                     Project                             recommendation
------------------------------------------------------------------------
Children's Health Fund, New York, NY for health                 100,000
 assessments, outreach, and education services for
 children and their families.........................
City of Laredo, TX for a community health assessment.           200,000
County of Marin, San Rafael, CA for research and                200,000
 analysis related to breast cancer incidence and
 mortality in the county and breast cancer screening.
Dillard University, New Orleans, LA for facilities              300,000
Iowa Chronic Care Consortium/Des Moines University,             200,000
 Des Moines, IA for a preventive health initiative...
Iowa State University, Ames, IA for facilities and            1,000,000
 equipment...........................................
Hope Institute for Children and Families,                       100,000
 Springfield, IL for facilities and equipment........
York College of Pennsylvania, York, PA for facilities           300,000
 and equipment.......................................
```

Figure 3: An allocation table.

If an earmark occurs in a table, the OMB does not actually cite the text of the table row the earmark appears in. Instead, they cite part of a bill, which alludes to the fact that an allocation is specified in a report accompanying the bill. This is the case in the example given in Table 1. Even when the excerpt is taken from the report containing the earmark, it will cite the section of the report that alludes to a table which contains the earmark; for example,

> Provided further, That within the amounts appropriated, $3,715,000 shall be used for the projects, and in the amounts, specified in the table titled "Congressionally-designated items" in the report of the Committee on Appropriations of the House of Representatives to accompany this Act. (Page 4 of HR bill 2847)

Thus, the excerpt is of little use in matching the table rows. It would, however, directly provide labels for earmarks that

[1] http://earmarks.omb.gov/earmarks-public/

[2] We use the Government Printing Office (GPO) as our source of congressional texts.

occur in plain text. As a result, we need to use the recipient and description fields for matching. In the example above, there is a perfect string match between the short description and the table cell corresponding to the 11th row of the project column. In general, the short description could be the concatenation of multiple table cells, a single table cell could be the concatenation of the short description and the full description, the description could be a permutation of entities in the table cell text, or the descriptions could contain abbreviations and misspellings of entities in the table cell text.

The matching task is even more complicated when recipients and descriptions are not unique within a document. A recipient can receive multiple earmarks within the same document, and multiple recipients can receive earmarks for the same purpose (e.g. “for equipment and facilities”). Furthermore, the same recipient can receive funding for the same purpose in multiple table rows within the same document. This means there is a one-to-many relationship between an earmark record from the OMB and table rows in a document.

Our matching evaluation works as follows. The matching algorithm is perfect if it matches a row for the earmarked appropriation in the document to its OMB record.

Consider a table row that represents an earmark. It will be incorrectly labeled negative if it does not get matched to a record from OMB either because the record is missing or because the matching was done incorrectly. It will correctly receive a positive label if it matches with a record from the OMB. Note that even if it was matched with the wrong OMB record, it would still be correctly labeled. So if we do matching for the purpose of labeling only, then for the algorithm to perfectly label the table rows, it must only match every table row that represents an earmark to some earmark.

Let’s now consider a row that does not represent an earmark. It will correctly receive a negative label if it does not get matched with a record from OMB. It will be incorrectly labeled positive if it matches with a record from OMB, either because the matching was done incorrectly or because the OMB contained a record that is not really an earmark.

For the purposes of this paper, we treat the database from OMB as definitional. We do not exert our own judgment in adding or removing earmarks from their records. We attempt to get the best labels we can by building the best matching algorithm that we can.

As mentioned above, multiple table rows can map to the same OMB record and mapping two OMB records to the same table row does not necessarily imply a labeling error. Because there are no clear constraints on the mapping from OMB records to table rows, we treat the matching problem as a simple classification problem.

5.4 Learning to Match: Training Data

As mentioned above, our goal is to build a classifier that, given an OMB record and a table row, predicts whether they match. We generated a training set to train this matching classifier in a semi-manual fashion. We first take an OMB record r and extract the set of documents D_r in which r occurs. Then for each document $d_r \in D_r$, we compute a similarity score $SIM(t_{d_r}, r)$ between every table row t_{d_r} and r. The similarity score is the maximum of the Jaccard similarities between the bigrams in t_r and the bigrams in r’s short description $r.sd$, full description $r.fd$ and recipient $r.rec$. For convenience lets define $F_r = \{r.sd, r.fd, r.rec\}$

$$JS(t_1, t_2) = \frac{|bigrams(t1) \cap bigrams(t2)|}{|bigrams(t1) \cup bigrams(t2)|} \qquad (1)$$

$$SIM(t_{d_r}, r) = \max_{f_r \in F_r} JS(t_{d_r}, f_r) \qquad (2)$$

Within each document, pairs of the OMB record and the table rows are ranked according to the similarity score SIM. We then looked at the top 20 pairs and hand label them as being a match or not. All other pairs are automatically labeled as not matching. Table 4 gives an example of descriptions from an OMB record and the five most similar table rows. Cells within the table are separated by the | symbol and are removed before computing similarities.

Short Desc	Trimble Local School District, Glouster, OH for an after-school program
Full Desc	NA
1	Trimble Local School District, Glouster, OH for an after-school program
2	Elementary and Secondary Education (includes FIE) | Trimble Local School District, Glouster, OH for an after-school program | Space
3	YMCA of Warren, Warren, OH for an after-school program
4	City of Newark, CA for an after-school program
5	Memphis City Schools, Memphis, TN for an after-school program

Table 4: Descriptions from an OMB record along with the top 5 most similar rows defined by SIM.

In this example, the earmark occurs in two tables within the same document, and the two corresponding rows are ranked highest. The other rows are also earmarks for after-school programs, but they are for different districts or organizations. We applied this labeling procedure to 516 randomly selected OMB records and found at least one matching table row 438 times, thereby giving 85% as an estimated lower bound of earmarks in tables. Because the OMB may cite multiple documents for every record, there were 840 cases in which we tried to match a record to a table row within a specific document, and we found at least one matching row 534 times. There are at least seven possible reasons an OMB record is not matched with a table row within a document that the OMB cites. For each error, assume the previous errors where not made:

1. The document is actually a bill that cites a report which contains the earmark.

2. The document is a public law or resolution; we do not include these documents in our analysis.

3. The citation was parsed incorrectly; we are looking in the wrong document.

4. The earmark does not appear in a table but in plain text.

5. The table was not parsed correctly, so the matching row is not available.
6. The matching row did not appear in the top 20.
7. The matching row was hand-labeled as not matching.

Although one might like to find every occurrence of every earmark, we are most concerned with finding every earmark at least once. The only issue is that if we fail to identify an earmark in a correctly parsed table, we would have an incorrectly labeled table row. This can only happen if errors 1-5 are not made and either error 6 or error 7 seven is made. If our matching algorithm is as good as the gold-standard human labels and the statistics above generalize, then we can estimate the upper bound of incorrectly labeled rows. We extracted 530k table rows and the OMB gives 122k occurrences of earmarks. In the worst case, where errors 1-5 are never made, we would have 8.4% of data mislabeled.

5.5 Learning to Match: Features

In this section, we describe the feature sets we designed for the matching task. Features are computed over pairs of table rows and OMB record pairs. For ease of notation, fix the OMB record r as well as the document d_r it appears in. Let T be the set of table rows t in d_r. Let F be defined as above as the set containing the short description, full description, and recipient texts from r. Let C_t be the set of table cells in table row t.

Jaccard Similarity Features:
Jaccard similarity between the table row and each field of the OMB record:

$$JS(t, f) \text{ for } f \in F \tag{3}$$

Maximum similarity between the table row and each field of the OMB record:

$$\max_{f_r \in F_r} JS(t, f_r) \tag{4}$$

Maximum Jaccard similarity between a field of an OMB record and each cell in the table row for each field of the OMB record:

$$\max_{c_t \in C_t} JS(c_t, f_r) \text{ for} f_r \in F_r \tag{5}$$

Maximum Jaccard similarity between all pairs of cells in the table row and fields of the OMB record:

$$\max_{(c,f) \in CxF} JS(c, f) \tag{6}$$

Relative Performance Features:
For any of the similarity features above, one can compare similarity scores for pairs of table rows and a particular OMB record within a document. A simple way to do this is to take the difference in similarity feature scores between a particular pair and the highest scoring pair. Alternatively, one can find the rank of a pair in the list of all pairs of table rows and a specific OMB record, where the order is determined by a similarity feature score. Here is an example of a difference feature:

$$SIM(t, r) - \max_{t' \in T} SIM(t', r) \tag{7}$$

Because the classification task is really a matching task, a particular OMB record will usually have only one matching table row per document. Providing information about how similar a table row is to an OMB record compared to others provides a way of normalizing similarity scores within the context of a particular OMB record and document.

5.6 Learning to Match: Experiments

As mentioned, the hand-labeling procedure was applied to 516 OMB records. It resulted in 769 matching pairs of OMB records and table rows and 647,157 non-matching pairs. There can be more matching pairs than OMB records because in some reports the same earmark can occur in two tables, resulting in two matches for a single record.

Out of the matching pairs, the lowest SIM score observed was 0.077. To reduce the number of negative examples, we only include pairs with SIM scores greater than 0.05. This reduced the number of negative instances to 32,715. One can think of this threshold as a high-recall, low-precision filter. Correspondingly, when we use the model to match all remaining pairs beyond the ones that were hand labeled, we compute SIM and label the pair negative if the value is less than 0.05. If the score is greater than 0.05, we use the model to label the pair. We present the result of our SVM classifier in Table 5. Varying the weight parameter on the sum of the slack variables in the SVM objective function in the range [0.001, 100] did not change performance.

	Mean Value (se)
AUC	0.9966 (0.08)
Precision: non-matching pairs	0.9203, (4.19)
Precision: matching pairs	0.9955, (0.26)
Recall: non-matching pairs	0.9373, (1.52)
Recall: matching pairs	0.9961, (0.12)
F-Score: non-matching pairs	0.9961, (0.12)
F-Score: matching pairs	0.9280, (2.02)

Table 5: Average Precision, Recall and F1 scores computed via 5 fold cross-validation.

To evaluate the quality of our features, we train a model using all the features described above as well as training a model on just the Jaccard similarity features, just the ranking features, and just the difference features. Figure 4 shows an ROC curve for each set of features. The ranking features are the best individual feature set, but including the difference features and the Jaccard similarity features gives a small but significant increase in the ROC AUC. The results show our algorithm can perform the matching task almost as well as an expert human annotator.

We applied the matching algorithm to all OMB records. For every OMB record r and for every document d referenced by r, we computed the features described above over pairs of table rows in d and r and record whether the matching algorithm predicts a match for each pair. Table 6 shows the number of OMB records, the number of OMB records that have at least one matching table row, and the percentage of OMB records that have at least one matching table row by year. The results for 2005 are dramatically worse, which we traced back to errors in linking documents in our corpus to documents that the OMB cites for earmarks in 2005.

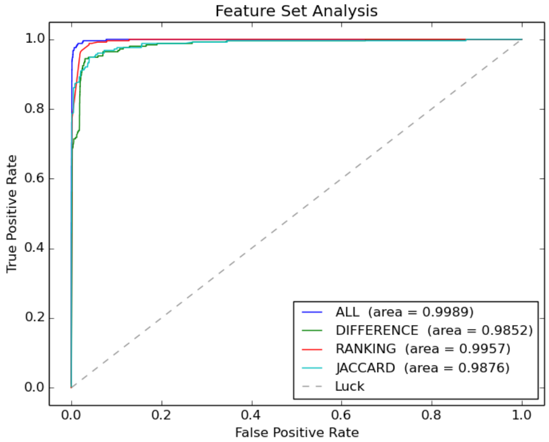


Figure 4: ROC curves for each feature set.

Enacted Year	OMB Records	Distinct	% Matched
2010	9785	9280	94.8 %
2009	11577	9181	79.8 %
2008	11503	10919	94.9 %
2005	14977	7624	50.1%

Table 6: Annual matching performance on OMB data.

6. EARMARK CLASSIFICATION

Given the labels on table rows induced by matching, we build a classifier that takes as input features computed over the table row and predicts earmark characteristics.

6.1 Earmark Classification Features

We compute four broad categories of features for the earmark classification task, which include geographic features, sponsor features, unigrams, and simple string heuristics.

Geo Features: presence of a city, presence of a county, and presence of a state

Sponsor Features: presence of a senator's last name and presence of a representative's last name

Unigrams: indicator variables for all unigrams in the training data except states, cities, counties, and last names of members of Congress.

Simple Heuristic Features:

- number of tokens
- number and percentage of tokens that are numbers
- number of percentage of tokens that are words
- number and percentage of characters that are of dots
- percentage of characters capitalized

6.2 Earmark Classification Experiments

To measure the generalization performance of our earmark classifier over time, we would like to train on documents from one year and test on documents from another. This is complicated by the fact that an earmark can occur in a document from the year it was enacted or the previous year. For example, when looking at the documents that the OMB references for earmarks enacted in 2009, we find that 3786 references are from documents dating from 2008 and 8500 references are dating from 2009. When grouping documents by year, we cannot use documents from 2010 since they could contain earmarks enacted in 2011, for which there is no OMB data. Our labeling policy is that an allocation is labeled as an earmark if and only if it matches an OMB record. Hence, we will mislabel all earmarks enacted in 2011, leading to poor training data and a poor classifier. We can however, use documents from 2008 and 2009.

Table 7 shows the results of training an SVM on the features described above. We are most interested in measuring how a model trained on one year performs on prior years, since most of the OMB data we need to fill in is from before 2008. We report cross-validated metrics for a model tuned and trained on 2009 documents. We also report the metrics for the 2009 model applied to documents from 2008. As a reference point for the generalization performance, we also include cross-validated metrics for a model tuned and trained on 2008 documents.

	CV 2009	**CV 2008**	**Train: 2009** **Test: 2008**
Precision	0.74 (0.017)	0.85 (0.014)	0.72
Recall	0.71 (0.042)	0.87 (0.017)	0.48
F-Score	0.72 (0.016)	0.86 (0.007)	0.58
ROC	0.93 (0.001)	0.97 (0.002)	0.91

Table 7: Metrics for documents grouped by document year.

The data suggest that if a document contains OMB earmarks, they were all enacted in the same year. Hence, we can assign an enacted year to those documents referenced by OMB earmarks. This allows us to group documents by the inferred enacted year and assign negative examples to the enacted year of the document they are in. This approach leaves out documents that are not referenced by any OMB records. Hence we may lose negative examples from omitted documents. The advantage is that we can increase our dataset by using documents enacted in 2008, 2009 and 2010. Table 8 shows results analogous to table 7.

	CV 10/09	**CV 2008**	**Train:10/09** **Test: 2008**
Precision	0.92(0.012)	0.90 (0.005)	0.87
Recall	0.93 (0.042)	0.93 (0.014)	0.84
F-Score	0.92 (0.016)	0.92 (0.005)	0.85
ROC	0.96 (0.004)	0.94 (0.004)	0.89

Table 8: Metrics for document grouped by enacted year.

Although there appears to be little difference in results between the two grouping approaches as measured by the area under the ROC curve, grouping by enacted year gives a much better F1. In both cases, cross-validation overestimates the generalization error, suggesting that locations, entities, and sponsors indicative of earmarks vary from year to year.

To evaluate the quality of our features, we train a model using all the features described above as well as training a

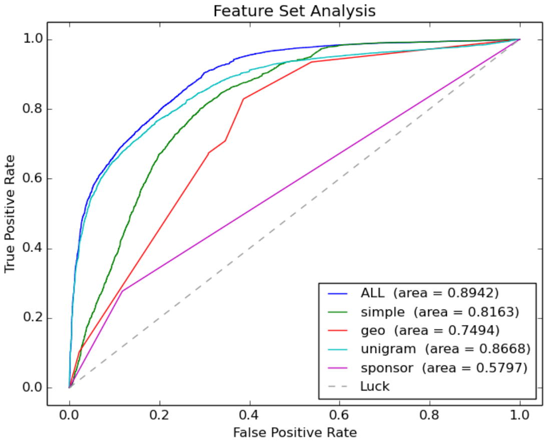


Figure 5: ROC curves for each feature set, trained on documents enacted in 2010 and 2009 and tested on documents enacted in 2008.

model on each set individually. Figure 5 shows an ROC curve for each feature set for a model trained on data enacted in 2009 and 2010 and tested on data enacted in 2008. Unigrams are the most powerful feature set, followed by the set of simple string heuristics.

7. A NEW EARMARKS DATASET

After retraining our model on all years, we applied our system, to documents going back to 1995. For each extracted allocation, we include:

Earmark Confidence Score: The score is the signed distance of the candidate earmark from the SVM margin. Positive scores reflect allocations predicted to be earmarks. The magnitude of the score corresponds to confidence in the prediction.

Allocation Location: We used OpenCalais, an off-the-shelf named-entity recognizer (NER), to geotag allocations. We obtained state-level locations for at least 85% of the earmarks and district-level associations for nearly 45% of the earmarks. Future work will include more sophisticated geotagging based on the location of entities mentioned in the text.

Allocation Topic: The original OMB data includes the spending committee associated with each earmark, such as Agriculture, Commerce, Education, Energy and Water, etc. We trained a spending committee classifier on the OMB data using a Softmax Regression. Before training, we collapsed spending committees related to Homeland Security, Military and Veterans Affairs, and Defense into a single category: Defense and Military Affairs. The average of the out-of-sample precision and recall scores for each class was approximately 85%. Using this classifier, we assigned spending committee labels to each allocation.

Figure 6 compares our generated dataset (DSSG) with the existing databases of earmarks from Citizens Against Government Waste (CAGW) and the Congressional Research Service (CRS). On average, our dataset includes approximately 3,000 more earmarks than CAGW and approximately 2,100 fewer earmarks than CRS. Our dataset also contains five times more earmarks in 2007 than CAGW. CAGW identified only 2,658 earmarks that year because of a “joint resolution that excluded pork from every appropriations bill except Defense and Homeland Security.”[5] Our system, however, identified budget items from the appropriation bills in 2007 that very closely resembled earmarked projects from other years.

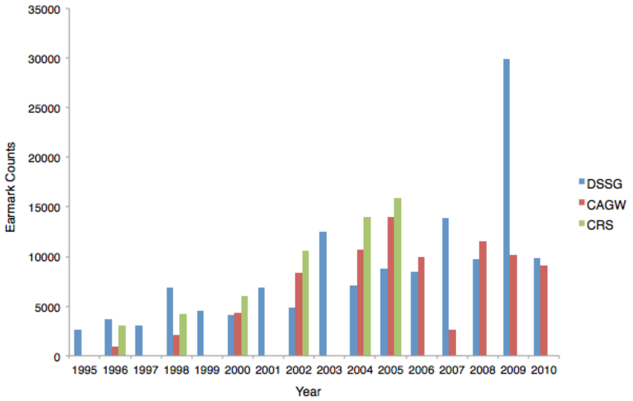


Figure 6: Comparison of our earmarks database (DSSG) with CAGW and CRS databases. Missing bars imply that the database doesn’t contain earmark counts for that particular year or those years couldn’t be retrieved easily.

The 2009 results look like an outlier, but we randomly examined 100 of those identified earmarks, and 95% of them were correct. We interviewed a K Street lobbyist, and he confirmed that these results are consistent with his impression of earmark behavior over the last decade. He said the big spike in 2009 earmarks is what led Republicans to ban earmarks in the House the following year [1]. Rather than finding too many earmarks in 2009, it may be that we found too few in other years.

From the 1990s through 2005, there is an upward trend in the number of earmarks in all three datasets. Then the use of earmarks appears to have declined except in 2009. The up-and-down trends in earmarks suggests a shift in the nature and processes of earmarking projects over time.

Using our dataset, we can conduct more longitudinal analyses of congressional processes. One outstanding question that has huge implications for political scientists and public policy is whether chairing an appropriations committee impacts the number earmarks granted to the chair’s state. Figure 7 shows the results of a difference-in-differences analysis for the nine times a chair of a House or Senate Appropriations committee changed hands between 1995 and 2010. Before a state gained the chair, it could be expected to have the same number of earmarks as other states that lacked the chair. But after a state gained the chair, it could expect about 15 more earmarks than the states that did not gain the chair—a 6% bump over the baseline. This relationship holds when leaving the 111th congress (which covered 2009 and 2010) out of the analysis.

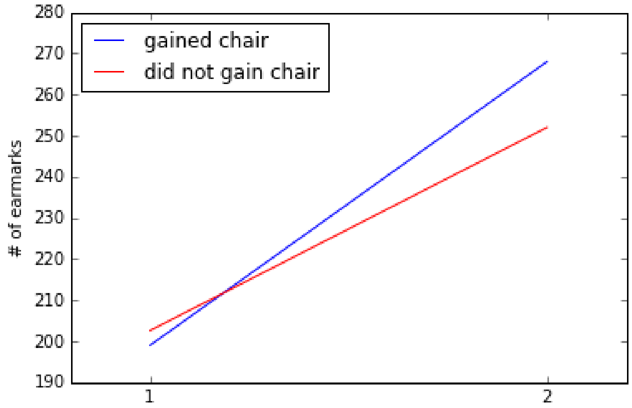


Figure 7: The number of expected earmarks before and after a state gained a chair of Appropriations (blue) and the number of expected earmarks before and after another state gained a chair of Appropriations.

8. FUTURE WORK

Our focus in this work has been on extracting earmarks from tables. We built the allocation table parser based on documents from 2005 and 2008-2010. If Congress used different formats in other years, the parser may fail to extract all allocations, leading to incomplete data. We plan to make a more in-depth survey of table formats used in congressional texts and generalize our table parser if necessary. Another avenue of future research involves identifying earmarks in free text. We believe the direct citations of free-text earmarks provided by the OMB make this task tractable. Finally, we hope to augment our earmarks dataset with more fields. In the current release, we provide the state and district the earmark went to if it is explicitly mentioned in the extracted allocation. We would like to provide more detailed geo-coding by locating the entities mentioned in the allocation. We also plan to include the dollar amount of the earmark, which involves determining the units the allocation is in.

9. CONCLUSION

It has been difficult to study how earmarking affects the legislative process due to the lack of comprehensive and open data on earmarks. This is mainly due to the immense amount of effort required by humans to sift through the thousands of pages of legal text produced by Congress each year. For the fiscal years 2005 and 2008–2010 the Office of Management and Budget (OMB) published a comprehensive dataset of earmarks through a massive inter-agency effort. We developed an automated system that learned how the OMB classifies budget allocations as earmarks. As a result, we were able to extend the scope of the OMB effort to additional fiscal years.

More generally, we have presented an example of how data mining and machine learning can be used to glean structured data from congressional documents. This structured data can be used to make transparent and quantitatively evaluate the functioning of legislative processes. Much of the difficulty in our work arose from the lack common table formats, document identifiers and document structure. These problems would be solved by Congress adopting machine-readable formats.

Our system is available at the Data Science for Social Good (DSSG) GitHub repository, and our dataset is available at the DSSG website: http://dssg.uchicago.edu/earmarks/. The dataset is already being used by Harris School of Public Policy researchers to do public policy research and analysis. In addition, it is being promoted by both the Center for Data Science and Public Policy as well as the Sunlight Foundation to researchers and practitioners interested in government transparency and public policy.

10. ACKNOWLEDGMENTS

We thank the Eric & Wendy Schmidt Data Science for Social Good Fellowship for generously supporting this work. We also thank Irina Matveeva and Rob Mitchum for useful discussions.

11. REFERENCES

[1] Anonymous Republican Lobbyist. Interview. Washington, D.C., October 30 2014.

[2] D. A. Austin and M. R. Levit. Mandatory spending since 1962. *Congressional Research Service*, March 23 2012.

[3] A. Bonica. Mapping the ideological marketplace. *American Journal of Political Science*, 58(2):367–386, 2014.

[4] S. Condon. Ron paul, don young, and joseph cao ignore gop earmark ban, risk reprimand. *CBS News*, April 2 2010.

[5] T. Finnigan. All about pork: The abuse of earmarks and the needed reforms. *Policy Briefing Series*, 2007.

[6] C. Hare and K. T. Poole. The polarization of contemporary american politics. *Polity*, 46(3):411–429, 2014.

[7] G. King. Replication, replication. *PS: Political Science & Politics*, 28(03):444–452, 1995.

[8] R. T. Meyers. *Strategic Budgeting*. Ann Arbor: University of Michigan, 1996.

[9] B. Montopoli. House republicans adopt earmarks ban in new congress. *CBS News*, November 18 2010.

[10] R. Nixon. Lawmakers finance pet projects without earmarks. *New York Times*, December 27 2010.

[11] D. Pinto, A. McCallum, X. Wei, and W. B. Croft. Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 235–242. ACM, 2003.

[12] R. Portman. Memorandum for the heads of departments and agencies. *Office of Management and Budget*, January 25 2007.

[13] P. Pyreddy and W. B. Croft. Tintin: A system for retrieval in text tables. In *Proceedings of the second ACM international conference on Digital libraries*, pages 193–200. ACM, 1997.

[14] B. Sinclair. *Unorthodox Lawmaking: New Legislative Processes in the US Congress*. CQ Press, 2011.

[15] S. Streeter. Earmarks and limitations in appropriations bills. *Congressional Research Service*, December 7 2004.

.