# PURPOSE OF THE PROJECT

The project for Litigation Operations, a company that reduces legal spend, used data to evaluate the quality of incoming law firm invoices. By using historical results of invoice non-compliance codes, an AI algorithm marked any non-compliant invoice entry. For example, an incoming invoice entry would be assigned a non-compensatory code for non-compliance work such as “Partners doing associate work” or “Vague invoice line-item description”.  Additional review by legal professionals was reduced by pre-populating the non-compensatory code.

An additional portion of the project included addressing generating benchmark topics for a new customer using NLP (Natural Language Processing) with Latent Dirichlet allocation (LDA)[[1]](#footnote-1),[[2]](#footnote-2). The topic provided a basis for the legal team to refine to finalize the topic using the frequently used words in open text responses. Figure 1 below illustrates my practicum journey with Litigation Operations.

Diagram

Description automatically generated

Figure 1. GaTech Litigation Operations Practicum Journey

## Legal Invoice Data Flow

Data for this project was from incoming litigation invoice line items from years 2021 through 2022. After the invoice is electronically received, AI predicts the task code, benchmark, activity, and non-compensatory code. After the AI assigns information, the legal team reviews the pre-populated results and corrects as needed. The QC Legal Reviewed corrected codes are used for the next day’s AI model rebuild. As a result, the system is continually learning and adjusting to any changes to contract compliance as shown in Figure 2.

An incorrect AI assigned task code can have a cascading error for the corresponding benchmark name, activity, and non-compensatory codes. This order is important because each following assignment uses the previous AI assigned information as shown in the flow in Figure 3.

Figure 2. Litigation Operations AI Process

**Assigned Non-Comp** Code

**Assigned Activities**

**Assigned Benchmarks**

**Corrected Task Codes**

Figure 3. AI Order of Assigned Information

## Text Analysis Methods

Graphical user interface

Description automatically generated with medium confidenceTo prepare the free-form litigation invoice description rows, for the AI algorithm, a series of text transformations were done to clean the data before topic extraction. First, the words were transformed to lowercase, and punctuation was removed so that the words could be equally counted without differentiating for different letter cases and interfering punctuation. Next, common words or stop words such as articles “the, a, of, an” and lower information words such as “and” and “does” were removed from the answers using the library sklearn.feature\_extraction.text. The remaining words in each response were ordered as single, double, and triple words (1-3 n-grams). The NLP pipeline is shown in Figure 4 below.

Figure .Natural Language Pipeline.

LDA only accepts input, x values, as a document word matrix. To add additional features, such as task code and timekeeper id, these were coded with a prefix and underscore to the clean line of text.

Using Chuang’s salient measure[[3]](#footnote-3) to highlight word extraction from the incorrect AI model invoice text item, the LDA models reveals important patterns useful to re-define individual attributes. The attributes and word entries are comingled so any salient relationship can be extracted. By removing extracted salient attribute ambiguity with better definitions, the AI models have more signal to use for training and increase accuracy with less need for manual quality control.

For a given word w, the conditional probability P(T|w) is the likelihood that observed word w was generated by latent topic T. The marginal probability P(T), which is the likelihood that any randomly selected word w’ was generated by topic T. Chuang defines the distinctiveness of word, w, as the Kullback-Leibler divergence[[4]](#footnote-4) between P(T|w) and P(T):

The distinctiveness weights the probability of the word in the corpus. So, if a word, denoted by w, occurs in all topics, observing the word does not add much information to the topic and would have a low distinctiveness score because log(1/1) = 0. The distinctiveness is balanced with the probability of finding the word. For example, a work that is only found once is a document would have a high distinctives score but its informational value is minimized because it only is found once which is reflected by the probability of finding the work P(w) as shown in the equation:

*saliency(w) = P(w) X distinctiveness(w)*

Saliency supports the identification of related words in a topic plus the differentiation between topics.

For LDA, a K value, or number of clusters or topics related to the question responses, needs to be known. The first cluster has the most information about the topic. The last cluster, K, contains the least amount of information. Too small of a K value results in clumping, where topics overlap. In contrast, too large a K value results in dissection, with insufficient meaning in the topics. Since the responses were relatively few compared to the generated number of words, the top cluster was manually reviewed between K values of 10 to 20, with 15 being optimal.

## Non-Compensatory Code Analysis

### Non-compensatory Code Metrics

Non-compensatory codes are assigned to an invoice row when the row may violate the law firm contract with the hiring company. Examples of a non-compensatory item are the line item may be too vague to describe the billed work done and when a highly paid law firm partner is doing that work that could be done by a lower pay rate legal associate. These are relatively infrequent events that happen less than 15% of the time for all entries in this data set. An infrequent event can have an uninformative accuracy. In this case, if no events are marked, the accuracy would be 85% with true negative accounting for all of the correct values.

Balanced accuracy[[5]](#footnote-5) normalizes the results by using the true positives and true negatives as shown below:

Recall can also be used as an indicator of model performance when it is more important that an item is marked instead of missed. For example, recall is used for medical testing when it is more important not to miss a diagnosis. Using the above example, the recall of returning no non-compensatory items would have a recall of 0%. High recall would mark items that are true positives.

### Non-Compensatory Codes Data Source

For the non-compensatory data set, Litigation Operations received 26,766 invoice line items between 12/1/2021 and 3/1/2022. Lawyers reviewed the AI predicted items and updated the non-compensatory codes. I gave analysis priority to the non-compensatory codes with the highest number of errors. For this data set, the counts for the greatest number of AI errors are shown in the Figure 4 below on the right. The data was grouped by non-compensatory code and error type. The range in number of errors is also shown in the box plot below in **Chart, box and whisker chart

Description automatically generated**Figure 4 on the left.

Figure . Box and Bar Charts of AI Non-Compensatory Error Counts

As a result of the most items AI incorrectly marked, I focused on “Vague” and “Partner doing Associate work” for optimizing the AI modeling parameters.

## New Benchmark Topics

Litigation Operations was on-boarding a new client. The client sent a sample of invoice entries for initial review. Because benchmarks are customized to a client, LDA topics were generated using the line-item entries. The legal team used the LDA extracted topics as a starting point for creating the final benchmark topics.

# ACCOMPLISHMENTS

## Unsupervised LDA Error Topics

Unsupervised learning using LDA topics was used to better understand the non-compensatory errors and generate benchmark topics. Topic extraction is a type of unsupervised learning that uncovers word clusters that are common between question responses as topics. David M. Blei developed a method to extract hidden or latent topics from a response called Latent Dirichlet allocation (LDA). LDA extracts meaningful word combinations from the user responses to summarize the results across all invoices. LDA is a well-known topic retrieval technique for documents. LDA was chosen for this project because of the support python library, pyLDAvis[[6]](#footnote-6), that generates easy to interpret topic charts.

For the non-compensatory reviewed items, attributes of QC reviewer name, ai assigned and final benchmark name, activity and task code were added to the text string to see if patterns existed across these items. The legal team members assigned an entry had their names obfuscated for this paper by using assigned letters.

### Vague Invoice Entries

For non-compensatory items, I generated LDA clusters for AI generated cascading errors from task codes, benchmarks, and activities that also missed vague entries.

Chart, bubble chart

Description automatically generatedThe first clustered topic, summarized by “email plaintiff trial review” and “deposition motion”, d\_b reviewed the activities of “Review Only”, “Communication Meetings” and AI incorrectly marked activity “Communications/Meetings”. This cluster accounted for 52% of all tokenized words for the missed entries shown in the PyLDAvis chart below in Figure 6.

Figure . LDA Vague Non-compensatory Topic Extraction

### Partners doing associate work

For non-compensatory items, I generated LDA clusters for AI generated cascading errors from task codes, benchmarks, and activities that also missed partners doing associate work non-compliant entries.

The first clustered topic, summarized by “review plaintiff with regard”, d\_h reviewed the activity of “Prepare/Prepare for” and activity “Communications/Meetings”. This cluster accounted for 51% of all tokenized words for the missed entries. This cluster is shown below in the PyLDavis chart in Figure 7.

Chart, bubble chart

Description automatically generated

Figure . LDA partners doing associate work non-compensatory topic extraction.

### Non-Compensatory Review

Business executives reviewed the cluster entries for consistency. LDA analysis showed unexpected results as it highlighted identical entries across legal matters as a cluster. Because these entries were duplicated up to 50 times, the frequencies increased the cluster location probability significantly. While this was not congruent with the goal of finding patterns to focus on quality improvement, it did emphasize an area for investigation.

An entry on the same date across many matters may indicate that a law firm might be over reporting hours. Further data was extracted to determine if a timekeeper was spending more than 12 hours per day and the rate that this work was being billed.

### New Benchmark Topics

Chart, bubble chart

Description automatically generatedBenchmarks were generated using line-item entries summarized from the LDA clusters for each of the new client’s task codes. Figure 8 below shows cluster one for task code B140. I extracted benchmark topic “Prepare/Draft/Confer Opposition to Stay” from looking at the cluster words.

Figure . Benchmark task code B140 topic extraction

# RESULTS

## Non-compensatory Code Analysis

The unsupervised analysis gave the most insight into areas where timekeepers where billing more than 12 hours and line items that were duplicated across matters. Both of these items have since been added as non-compensatory codes that are marked on import.

## New Benchmark Topics

New client LDA topics were generated using line-item entries summarized from the LDA clusters. Legal used LDA topics as a starting point for creating the final benchmark topics. LDA topic extraction accelerated legal final topic creation from a limited number of entries. The results are shown below in Table 1 including the legal team’s assigned final topic.



Table 1. Benchmark LDA Topic Results

# Bibliography and Credits

1. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. J. Mach. Learn. Res. 2003, 3, 993–1022.
2. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. In Advances in Neural Information Processing Systems; University of California: Berkeley, CA, USA, 2002; pp. 601–608.
3. Chuang, Manning, C., & Heer, J. (2012). [Termite. Proceedings of the International Working Conference on Advanced Visual Interfaces](http://vis.stanford.edu/files/2012-Termite-AVI.pdf), 74–77. <https://doi.org/10.1145/2254556.2254572>
4. S. Kullback and R. A. Leibler. On information and sufficiency.Annals of Mathematical Statistics, 22(1):79-86, 1951.
5. Mower, Jeffrey P. (2005-04-12). "PREP-Mt: predictive RNA editor for plant mitochondrial genes". BMC Bioinformatics. 6: 96. doi:10.1186/1471-2105-6-96. ISSN 1471-2105. PMC 1087475. PMID 15826309.
6. Sievert, Carson & Shirley, Kenneth. (2014). [LDAvis: A method for visualizing and interpreting topics](https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf). 10.13140/2.1.1394.3043.

1. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet allocation. J. Mach. Learn. Res. 2003, 3, 993–1022. [↑](#footnote-ref-1)
2. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet allocation. In Advances in Neural Information Processing Systems; University of California: Berkeley, CA, USA, 2002; pp. 601–608. [↑](#footnote-ref-2)
3. Chuang, Manning, C., & Heer, J. (2012). [Termite. Proceedings of the International Working Conference on Advanced Visual Interfaces](http://vis.stanford.edu/files/2012-Termite-AVI.pdf), 74–77. https://doi.org/10.1145/2254556.2254572 [↑](#footnote-ref-3)
4. S. Kullback and R. A. Leibler. On information and sufficiency.Annals of Mathematical Statistics, 22(1):79-86, 1951. [↑](#footnote-ref-4)
5. Mower, Jeffrey P. (2005-04-12). "PREP-Mt: predictive RNA editor for plant mitochondrial genes". BMC Bioinformatics. 6: 96. doi:10.1186/1471-2105-6-96. ISSN 1471-2105. PMC 1087475. PMID 15826309. [↑](#footnote-ref-5)
6. Sievert, Carson & Shirley, Kenneth. (2014). [LDAvis: A method for visualizing and interpreting topics](https://nlp.stanford.edu/events/illvi2014/papers/sievert-illvi2014.pdf). 10.13140/2.1.1394.3043. [↑](#footnote-ref-6)