Text Complexity and Readability

Final Project Definition

By:

Yifat Gastfraind  
ID: 043139419  
Email: yifatb18@gmail.com  
Phone: 052-3124611

Tomer Surujon  
ID: 318161767  
Email: tomer.surujon@gmail.com  
Phone: 052-2292110

Revital Yusupov  
ID: 322355264  
Email: revital2409@gmail.com  
Phone: 054-6727722

Ori Barel  
ID: 207813007  
Email: oribarel80@gmail.com  
Phone: 054-7981113

Guy Farbstein  
ID: 205884927  
Email: guymkc7@gmail.com  
Phone: 050-6310001

Supervisor:

Dr. Or Anidjar

Git:

[Link to the Git Repository](https://github.com/YifatGast/Final-Project.git)

**Table of Contents**

[1. Introduction 3](#_Toc186411647)

[1.1. Contribution 4](#_Toc186411648)

[2. Related Work 5](#_Toc186411649)

[3. Functional Description and Requirements 7](#_Toc186411650)

[3.1. Dataset 8](#_Toc186411651)

[3.2. Data Ingestion 9](#_Toc186411652)

[3.3. Preprocessing 9](#_Toc186411653)

[3.4. Embedding, Similarity Function and Analysis Execution 9](#_Toc186411654)

[3.5. Evaluation and Comparison 10](#_Toc186411655)

[3.6. Visualization 10](#_Toc186411656)

[3.7. Reporting 11](#_Toc186411657)

[4. Architecture 11](#_Toc186411658)

[4.1. Components Overview 11](#_Toc186411659)

[4.2. Client Side 11](#_Toc186411660)

[4.3. Server Side 11](#_Toc186411661)

[4.4. Database 11](#_Toc186411662)

[4.5. External Resources 12](#_Toc186411663)

[5. Work plan 12](#_Toc186411664)

[6. Client side 13](#_Toc186411665)

[6.1. Usage Illustration: 13](#_Toc186411666)

[6.2. Mockup 14](#_Toc186411667)

[7. Server Side 17](#_Toc186411668)

[7.1. API 17](#_Toc186411669)

[8. References 17](#_Toc186411670)

# Introduction

Laws, and particularly tax laws, represent complex, detailed, and multi-layered legal texts that often pose challenges in understanding, interpretation, and navigation. This complexity is not limited to legal professionals but also affects consultants and ordinary citizens alike. Despite frequent efforts by lawmakers to simplify legislative texts and the regulations derived from them, aiming to make them more accessible and comprehensible, reality often demonstrates otherwise. In many cases, reforms and legislative changes result not in simplification but in further complications of the texts, adding nested structures, conflicting provisions, and intricate language that make the laws more challenging to use and apply effectively. The U.S. Tax Code serves as an excellent example of the aforementioned challenges.

The U.S. Tax Code represents one of the most complex legal texts, with deeply nested structures, extensive cross-references, and specialized language that often make it inaccessible to the average reader. This complexity leads to frequent misinterpretations, higher compliance costs, and inefficiencies for both taxpayers and policymakers.

The Tax Cuts and Jobs Act (TCJA) of 2018 introduced significant changes to the Tax Code, which were ostensibly intended, among other goals, to simplify the tax laws that preceded it. However, some argue that the tax reform has resulted in even greater complexity, creating substantial challenges in understanding the law, analyzing it, and assessing its implications for taxation in the U.S. and its broader economic impacts. The TCJA created a unique opportunity to analyze and evaluate the structural and linguistic changes brought about by such legislative modifications. This study aims to examine the impacts of the TCJA, providing insights into the readability and usability of legal texts, with the ultimate goal of influencing future legislative drafting.

Existing tools such as LexisNexis[[1]](#footnote-1) and readability metrics like Flesch-Kincaid[[2]](#footnote-2), provide only surface-level evaluations, failing to capture the intricate relationships between sections or their hierarchical dependencies. Similarly, modern NLP models like OpenAI GPT and Hugging Face Transformers[[3]](#footnote-3) excel in general text processing, but require extensive customization to handle the unique demands of legal documents, particularly for tasks, such as cross-referencing and hierarchical analysis.

This study leverages advanced natural language processing techniques, such as the Language-agnostic BERT (LaBSE), to bridge these gaps. The methodology for analyzing and comparing text complexity and readability focuses on extracting and structuring data from legal texts, mapping their hierarchical relationships, and integrating semantic and structural metrics into a unified framework. Challenges such as data extraction from PDFs, hierarchical parsing, and combining multiple complexity metrics, are addressed to ensure meaningful and actionable insights.

## Contribution

The societal impact of this work is significant. This project introduces an innovative framework for analyzing legislative complexity, combining structural and semantic metrics into an integrated. It provides a detailed assessment of legal texts that goes beyond traditional readability metrics. Simplifying and quantifying legislative complexity improves accessibility, fosters compliance, and reduces administrative burdens. By providing policymakers with detailed analyses of the TCJA, this study not only enhances understanding of the law, but also establishes a framework for analyzing and improving legal texts across various domains. The study also demonstrates the practical implications of these insights, offering a roadmap for drafting more accessible laws. In doing so, it bridges a critical gap in the field, providing stakeholders with tools to better understand and improve the usability of complex legal documents.

# Related Work

## NLP and Legal platforms and tools

The complexity of analyzing laws and regulation, such as tax codes, is a pressing issue that intersects technical, legal, and societal domains. Despite numerous tools and advancements in natural language processing (NLP), existing systems fall short of addressing the intricacies of legislative texts.

One widely used solution is LexisNexis - a robust legal research platform that provides access to a broad array of documents. However, it lacks the analytical capabilities required for mapping structural dependencies and comparing hierarchical elements in legislative texts. Similarly, tools like CaseText[[4]](#footnote-4) focus primarily on document management and search functionalities, providing little to no insight into text complexity or readability metrics.

Readability metrics, such as Flesch-Kincaid and SMOG[[5]](#footnote-5), have historically been applied to analyze textual accessibility. While these tools offer surface-level assessments, they are not designed to capture layered structures, such as nested clauses and cross-references, inherent in legal texts like the U.S. Tax Code. The inability to integrate semantic and structural analysis limits their utility for legislative applications.

Recent advancements in NLP have introduced more sophisticated models like OpenAI GPT and Hugging Face Transformers, including BERT and LaBSE (Language-agnostic BERT). These models excel in tasks such as summarization, sentiment analysis, and sentence embedding. However, their application to legal texts requires significant customization. For instance, while LaBSE generates high-quality semantic embeddings, it does not natively address the hierarchical structures or cross-referential complexities that define legislative documents. Similarly, OpenAI GPT excels in semantic analysis, but lacks tailored tools for quantitative comparisons across versions of legal texts.

Emerging technologies like Spacy LegalNLP[[6]](#footnote-6) and AI-powered OCR[[7]](#footnote-7) tools address domain-specific challenges. Spacy LegalNLP enables entity recognition and tokenization tailored to legal contexts, but does not provide comprehensive metrics for readability or structural mapping. AI-powered OCR tools effectively extract textual data from unstructured formats, such as PDFs, but often fail to preserve the hierarchical relationships critical for analysis.

Our approach integrates these advanced technologies with custom-designed methodologies to address their limitations. By combining LaBSE embeddings with hierarchical mapping algorithms, we create a unified framework for analyzing legal texts. This system enables:

* **Comparative Analysis**: Highlighting structural, semantic and lexical differences between pre- and post-TCJA texts.
* **Semantic and Readability Metrics**: Offering actionable insights into textual complexity.
* **Hierarchical Parsing**: Analyzing nested structures, sections, and subsections with precision.

In doing so, our solution bridges critical gaps in existing tools and methodologies, delivering a comprehensive platform tailored to the unique challenges of legal text analysis.

## Complexity of U.S. Tax Code

The U.S tax code has faced extensive criticism over the years for being excessively lengthy and complex. This argument was particularly pronounced during the tax reform of 2017, with the enactment of the Tax Cuts and Jobs Act (TCJA). Given the complexity and lack of clarity of the tax code, and the direct connection between taxation and other economic and legal fields, studies are conducted periodically to provide insights and practical implications for taxpayers and policymakers.

A comprehensive article discussing the complexity of the U.S tax code between 1997 and 2017, through textual analysis, was published in 2024[[8]](#footnote-8). The article presented various metrics used by the researchers to determine the complexity of the tax code, including the number of nodes, the frequency of cross-references to other sections, and the prevalence of rare terms within the code. These metrics were examined on a sample of chapters and sections of the tax code. However, the study did not analyze the level of complexity before and after the tax reform, which, as noted, some argue exacerbated the opacity and complexity of the text. Additionally, no natural language processing models were employed to examine semantic similarity and linguistic patterns between the texts, analyze the semantic relationships and connections between chapters and expressions, or compare and assess the hierarchical structure of the tax code.

This project aims to integrate the aforementioned metrics and develop a comprehensive, holistic, and integrative measure for analyzing text complexity and comparing texts.

# Functional Description and Requirements

The system is designed to analyze and compare complexity metrics for various legal and regulatory texts, presenting insights through visualizations and detailed reports. Each component is designed to ensure seamless functionality and adaptability for diverse user needs.

**Key Features**

The system incorporates several critical features to facilitate comprehensive analysis and actionable insights:

1. **Semantic and Structural Analysis:** Seamlessly integrates advanced embeddings with complexity metrics, enabling detailed evaluations of both readability and content structure.
2. **Hierarchical Parsing:** Ensures precise mapping of nested structures within the U.S. Tax Code, preserving dependencies between sections and subsections for a deeper understanding of the text's organization.
3. **Customizable Outputs:** Provides stakeholders with the ability to tailor visualizations and reports according to their specific requirements.

These features collectively establish a robust foundation for analyzing and interpreting legislative texts with unparalleled precision.

**General Functional Requirements**

Before delving into the detailed components of the system, the following is a general list of its functional requirements:

* **Document Upload and Parsing:** Ability to upload and parse documents in various formats (e.g., PDF, DOCX).
* **Complexity Metrics Computation:** Calculation of complexity metrics, such as semantic and lexical similarity, branching, nested conditions, and more.
* **Inter-Textual Analysis:** Capability to identify structural and semantic differences between texts.
* **Visualization and Reporting:** Provision of detailed reports and graphical visualizations.
* **Export Functionality:** Ability to export results in multiple formats (e.g., PDF, CSV).

## Dataset

The dataset for this study is derived from publicly available versions of the U.S. Tax Code for the years 2017 and 2018, ensuring relevance to the analysis of pre- and post-TCJA reforms. As part of the overall system, code was developed and implemented to extract relevant and essential information from PDF documents. Within the algorithm, headings, titles, and sections were identified and extracted, both at the name and content levels, all while preserving the original hierarchical structure of the documents for comparison purposes. The data organized into structured CSV files for detailed analysis, with the following steps:

* Text Extraction: All headings, titles, sections and paragraphs are identified and extracted hierarchically from each Tax Code (2017 and 2018).
* Data Cleaning: Unnecessary elements, such as formatting artifacts are removed, while semantic and structural integrity is preserved.
* Structured Organization: Data is reformatted into analyzable formats, while preserving the hierarchical relationships and structural connections between the sections.
* Database Storage: The processed dataset is stored in a new database for efficient retrieval and analysis.

## Data Ingestion

The system accepts various documents as input and converts them into structured formats that can be analyzed and compared. It supports various document/file types, such as PDF, DOCX, and others, to accommodate a wide range of data sources, making it versatile and user-friendly.

## Preprocessing

During preprocessing, the system cleans and tokenizes the data, ensuring that irrelevant elements are removed. It also parses hierarchical structures, enabling the identification and preservation of relationships between sections and subsections for detailed analysis.

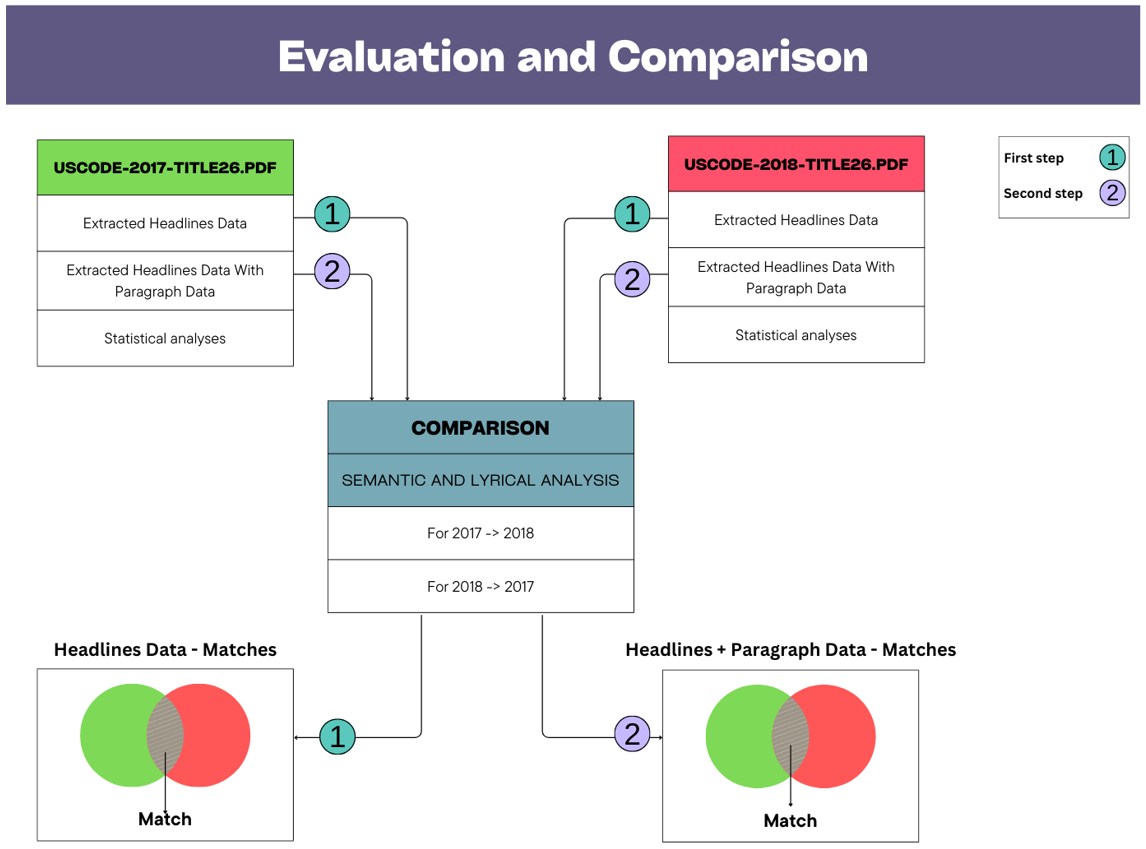
## Embedding, Similarity Function and Analysis Execution

Central to the framework are sentence and content embeddings, generated using Language-agnostic BERT (LaBSE). These embeddings capture semantic and contextual relationships, forming the basis for evaluating semantic similarities between different texts. To evaluate similarities, the system uses a combination of metrics, including semantic similarity (cosine similarity on embeddings), lexical similarity (e.g., Jaccard Index and Levenshtein distance), and syntactic measures. Paragraphs with high similarity scores are further cross-compared in a Cartesian product, enabling detailed insights into the structural and linguistic changes between the pre- and post-TCJA Tax Code versions.

Additionally, to create an integrative complexity metric, additional aspects were examined to provide insights into the complexity and readability of each text, such as analyzing the frequency and other statistical measures of words and phrases appearing in each version of the Tax Code - before and after the reform, semantic annotations, cross-references, and additional linguistic features identified through advanced natural language processing techniques, such as N-Grams.

## Evaluation and Comparison

The evaluation framework includes a range of metrics to assess changes in readability, semantics, and structure. Key performance metrics include changes in readability (e.g., Flesch-Kincaid scores), semantic shifts identified through embeddings, and lexical changes. Structural metrics focus on nested clauses, cross-references, and hierarchical depth. Comparative visualizations, such as graphs and heatmaps, illustrate the differences between tax code versions, providing stakeholders with actionable insights to understand the impacts of the TCJA reforms.



## Visualization

The system features interactive dashboards that enable users to explore the results of text complexity analysis and comparisons in depth, deriving insights through various visual tools, including graphs and heatmaps. These visualizations are customizable, enabling users to filter and compare results dynamically, according to their specific needs.

## Reporting

The reporting module generates detailed, exportable reports that include critical metrics, such as readability and complexity scores, branching, nested conditions, internal cross-reference and hierarchical mappings. These reports are available in various formats, including PDF and Excel, ensuring accessibility and flexibility for a range of stakeholders. This comprehensive system framework ensures a robust and adaptable solution for analyzing complex legal documents.

# Architecture

## Components Overview

* **Client Side:** A web-based user interface (UI) for document uploads, configuration, and results visualization.
* **Server Side:** Backend services to process, analyze, and compute metrics using NLP pipelines.
* **Database:** Storage for processed documents and analysis results.
* **External Resources:** Pretrained NLP models (e.g., BERT, GPT), libraries (e.g., pandas, matplotlib), and cloud infrastructure for computation.

## Client Side

* Interface for uploading documents.
* Visual display of complexity metrics and comparisons.

See further details and elaboration in Section 6 below.

## Server Side

* Document parsing and preprocessing.
* NLP-based analysis modules.
* APIs for interaction with the client-side UI.

See further details and elaboration in Section 7 below.

## Database

* Stores uploaded documents and their processed metadata.
* Indexed for efficient retrieval.

## External Resources

* Language-agnostic BERT (LaBSE) and Hugging Face Transformers for semantic analysis.
* Python libraries for data processing and visualization.

# Work plan

**Milestones and Timeline:**

1. Data Collection and Preprocessing (Weeks 1–5):

* Download publicly available versions of the U.S. Tax Code for the years 2017 and 2018 in PDFs
* Extract data and features (building script) from the Tax Codes
* Clean and structure the data into CSV format
* Fine-tune extracted data and features according to initial analysis results and additional requirements

1. Model Integration (Weeks 3–9):

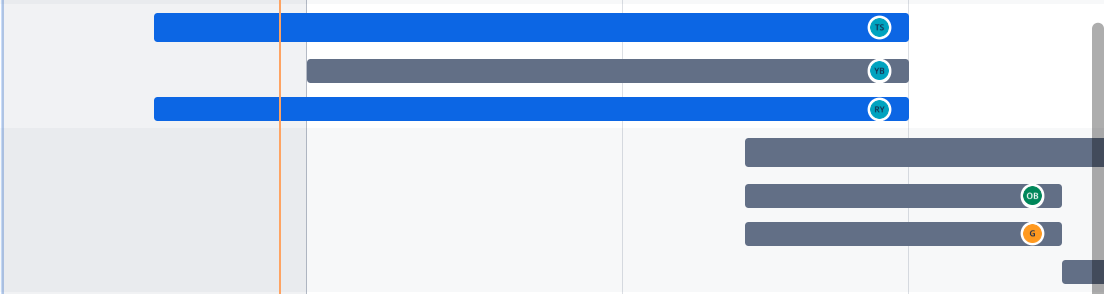
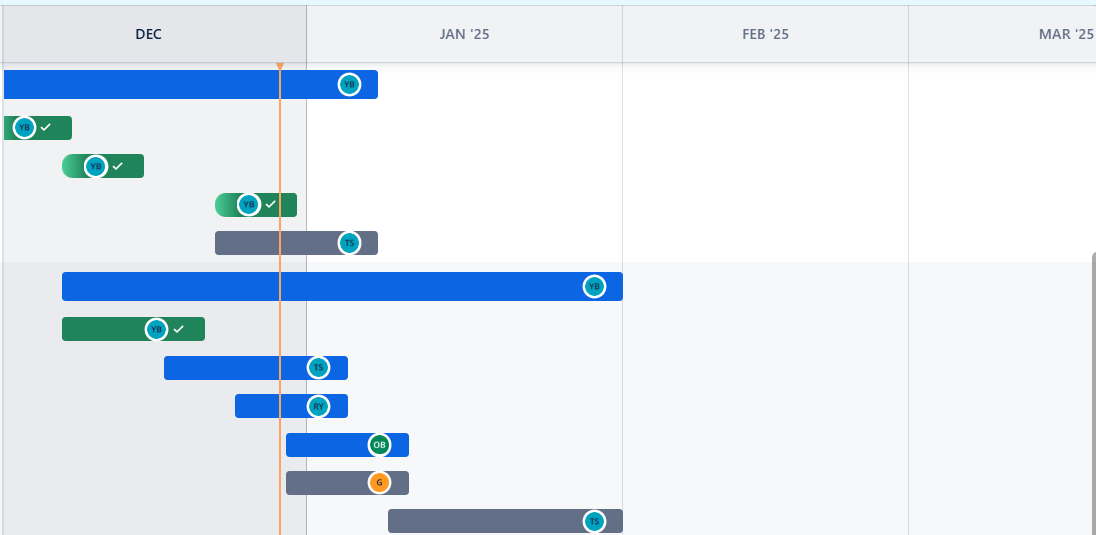
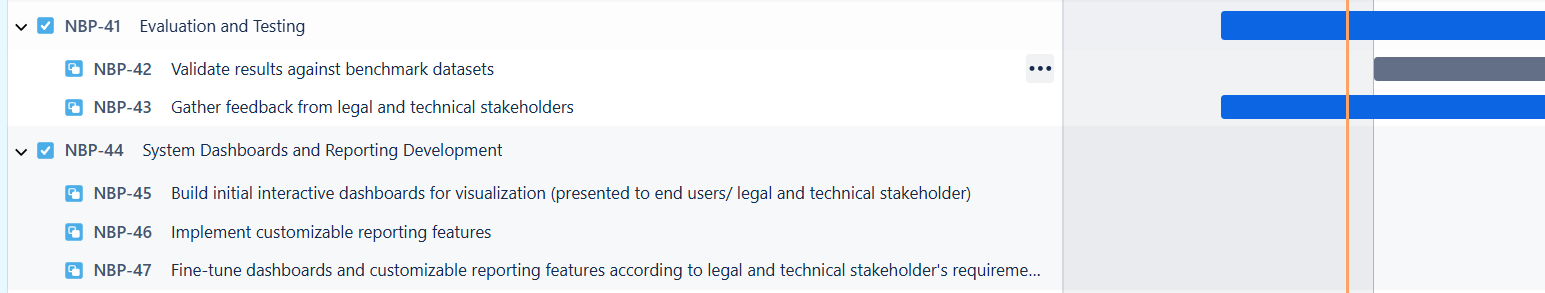
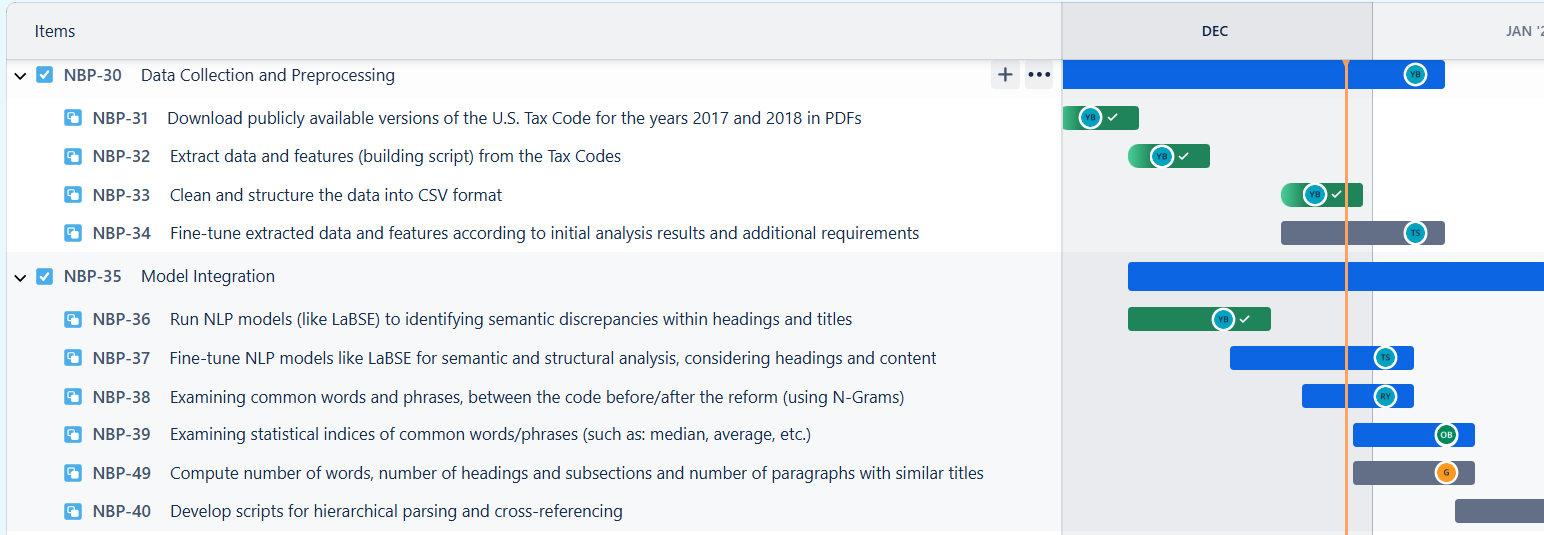
* Run NLP models (like LaBSE) to identifying semantic discrepancies within headings and titles
* Fine-tune NLP models like LaBSE for semantic and structural analysis, considering headings and content
* Examining common words and phrases, between the code before/after the reform (using N-Grams)
* Examining statistical indices of common words/phrases (such as: median, average, etc.)
* Compute number of words, number of headings and subsections and number of paragraphs with similar titles
* Develop scripts for hierarchical parsing and cross-referencing

1. Evaluation and Testing (Weeks 13–16):

* Validate results against benchmark datasets
* Gather feedback from legal and technical stakeholders

1. System Dashboards and Reporting Development (Weeks 9–12):

* Build initial interactive dashboards for visualization (presented to end users/ legal and technical stakeholder)
* Implement customizable reporting features
* Fine-tune dashboards and customizable reporting features according to legal and technical stakeholder's requirements



# Client side

The client side of the system is designed to prioritize usability and accessibility for a wide range of users, including policymakers, legal professionals, and analysts.

## Usage Illustration:

1. Scenario 1: A user uploads two versions of a regulatory text and views a comparison of their complexity and readability metrics.
2. Scenario 2: A user analyzes a single document to extract insights about its structural complexity.
3. Scenario 3: A user generates a comprehensive report with visualizations for a presentation.

## Mockup

The client-side interface includes:

**Frame 1**: Home Page

The home page introduces the platform's purpose—providing actionable insights into text complexity and readability. It highlights the platform's features, including blog access, company history, and contact support. The design invites users to explore and start their comparative analysis journey.

תמונה שמכילה טקסט, צילום מסך, מולטימדיה, תוכנה

התיאור נוצר באופן אוטומטי

**Frame 2**: Upload Documents

This screen serves as the starting point for comparing two documents. Users can drag and drop or browse files to upload. Supported formats include PDF, Word, and plain text. The page emphasizes simplicity and versatility, enabling users to initiate analysis with just a few clicks.

תמונה שמכילה טקסט, צילום מסך, תוכנה, גופן

התיאור נוצר באופן אוטומטי

**Frame 3**: Comparative Dashboard

This interactive dashboard showcases side-by-side comparisons of two documents. Key metrics, such as total word count, unique word count, and headings, are prominently displayed. Visualizations like scatter plots, bar charts, and pie charts provide insights into cross-references, semantic differences, and structural complexities, helping users identify critical changes.



# Server Side

The server-side architecture is built to handle large datasets efficiently and perform complex analyses, ensuring seamless integration with the client interface.

## API

The system includes a RESTful API that facilitates communication between the client and server, supporting key functionalities:

* POST /upload: Accepts document uploads for analysis.
* GET /analyze: Returns computed metrics for a document.
* POST /compare: Compares two documents and returns results.
* GET /export: Exports analysis results in the desired format.

# References

1. **LexisNexis tools documentation**: Explains the capabilities and limitations of LexisNexis as a legal research tool.
2. **BERT and GPT model documentation (Hugging Face)**: Covers the use of transformer models for NLP tasks and their applicability to text analysis.
3. **Readability Test Tool descriptions**: Discusses the algorithms and metrics used for measuring readability in textual data.
4. **"Attention Is All You Need" (Vaswani et al., 2017)**: The foundational paper on Transformer architecture, forming the backbone of BERT and GPT models.
5. **"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" (Devlin et al., 2019)**: Explores the development and applications of BERT for NLP.
6. **"Measuring Readability in Natural Language Processing" (Collins-Thompson, 2014)**: A comprehensive review of readability metrics and their applications in text analysis.
7. **"Natural Language Processing with Transformers" (Lewis et al., 2020)**: Highlights the evolution of transformer models and their implications for NLP tasks.
8. **"Deep Learning for Text Analysis: A Review" (Young et al., 2018)**: Discusses advancements in deep learning for text analysis, including readability and semantic evaluation.

1. LexisNexis is a leading legal research platform used by lawyers, researchers, and professionals in regulatory and legal fields. The system provides access to rich databases that include laws, court rulings, contracts, academic articles, and regulatory documents. [↑](#footnote-ref-1)
2. The Flesch-Kincaid metric is a tool used to evaluate the readability of texts, determining how easy or difficult a text is to understand. It was developed by Rudolf Flesch and J. Peter Kincaid and is widely used in fields such as education, advertising, and law. The metric is based on the structure of the text, taking into account sentence length and word complexity. [↑](#footnote-ref-2)
3. Hugging Face Transformers is a popular and advanced open-source library that provides access to natural language processing (NLP) models and advanced transformer models, such as BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer), LaBSE (Language-agnostic BERT Sentence Embedding), etc. [↑](#footnote-ref-3)
4. CaseText is an AI-powered tool primarily used by legal professionals for efficient legal research. The platform provides access to rich databases of court rulings, laws, regulations, and other legal materials, utilizing innovative technologies that enable quick identification of relevant information.  
   CaseText is particularly known for its CoCounsel tool, an AI-based legal assistant that allows the execution of complex legal tasks quickly and accurately. [↑](#footnote-ref-4)
5. SMOG (Simple Measure of Gobbledygook) is a readability metric that evaluates the difficulty level of a text and the educational level required to understand it. The metric is based on counting the number of words with three or more syllables in 30 sentences from the text (10 sentences from the beginning, middle, and end), and then calculating the readability level using a simple formula. [↑](#footnote-ref-5)
6. Spacy LegalNLP is an extension of the popular natural language processing (NLP) library Spacy, specifically tailored for analyzing legal texts. The tool is designed to assist in identifying and analyzing elements unique to legal documents, such as laws, contracts, and rulings. [↑](#footnote-ref-6)
7. OCR (Optical Character Recognition) is a technology that converts text from images or scanned documents into digital text that can be edited, searched, and analyzed. OCR is used for a wide range of applications, from document processing to extracting text from images. [↑](#footnote-ref-7)
8. Complexity and the Tax Code: A Textual Analysis, By Charles Swenson, HaoQu and Xiao Song (Public Finance Review 2024, Vol. 52(4)). [↑](#footnote-ref-8)