Covert Linguistic Annotation for Legal Document Abstraction

BrokerSavant Inc ©
www.brokersavant.com

Abstract

Legal document abstraction is a common business task that involves identifying key points in contracts used for various financial and legal purposes. The abstraction process is a laborious task performed by trained legal professionals that is particularly well suited for automation via Machine Learning. The abstraction workflow involves linguistic annotation (bookmarking or highlighting of relevant sections). If appropriately captured, this linguistically annotated text can be used as training data for supervised Named Entity Recognition and Text Classification. This demonstration presents an online system used for commercial lease abstraction that captures linguistically annotated data.

1 Introduction

Legal document abstraction is the process of identifying key pieces of information in various contracts, e.g., commercial real estate (CRE) leases, and related legal documents. In particular, a lease abstract is a summary of the key financial, business, and legal information that exists in a CRE lease. Lease abstracts are needed, for example, for financial statements and audits, lease administration, and portfolio management. The abstracted information is typically entered in structured form in a spreadsheet, database, lease administration, or financial software.

The lease information that should be abstracted depends on the reasons for abstracting, and the length and complexity of the lease documents. Lease abstracts can range from several to hundreds of fields per lease. At a minimum, an abstract should contain the most important dates and dollars provisions of the lease.

In a typical workflow, lease abstraction begins by a legal professional (typically a paralegal) collecting all documents relevant to the leased properties (leases, lease amendments, work letters, etc.). Once collected, the task of the abstractor is to review all documents in detail, identify all sections relevant for a particular field, bookmark them, and provide a draft abstract. As lease abstraction errors are associated with high liability, documents and lease abstracts are typically reviewed by multiple abstractors and signed off by a senior member of the abstraction team, usually an attorney.

Executed contracts are typically reviewed in a digital form (e.g., scanned documents in PDF format) with a document viewer. The abstracted fields are then entered in a separate application. This workflow places additional burden on abstractors who need to review lengthy and complex documents with great attention to detail. The average size of an office lease is 60 pages, with office leases occasionally containing more than 250 pages. Consistent document formats are not common as they largely depend on the attorneys, brokers, and companies involved. Documents are often scanned and contain image content which precludes the convenience of text-based document search.

2 Motivation

CRE lease abstraction is typically performed by trained legal professionals. It is a laborious process with an average review time ranging from 2 to 8 man-hours and cost that averages $250 per
lease. Semi-automating the lease abstraction process could significantly reduce document review time and result in major cost savings.

The lease abstraction task is particularly suitable for the application of Machine Learning (ML) techniques for two main reasons. First, identifying text relevant to a particular abstraction field can be performed utilizing supervised Named Entity Recognition and Text Classification techniques. Second, and more importantly, the task can be performed utilizing ML techniques without incurring the cost of obtaining dedicated training data. The process of lease abstraction involves highlighting and bookmarking text relevant to a set of abstraction fields, which is, in essence, a linguistic annotation task. Harnessing the created highlights and bookmarks as training data can result in an inexpensive (and thus practical) ML system for semi-automating the lease abstraction workflow.

3 System Overview

We have developed an online system that serves a dual purpose. It allows lease abstractors to review and abstract legal documents following their normal workflow. At the same time, the system is collecting linguistically annotated data that can in turn be used for ML purposes. Figure 1 shows an overview of the system. Executed legal documents (typically scanned images in PDF format) are submitted and processed by Optical Character Recognition (OCR) software\(^2\). The OCR output is converted to an HTML format in which the original scans (images) are seen and overlaid with an invisible text layer (the corresponding OCR-ed text). The leases are then reviewed by legal abstractors following their typical workflow. The manually created data is used to both produce a final abstract artifact, as well as create training data for ML purposes.

The system requirements involve preserving a typical abstraction workflow while at the same time collecting linguistically annotated text. The system needs to be secure, as it handles confidential information. The original document scan (with formatting, tables, mark-up, signatures, etc.) needs to be displayed while at the same time textual data needs to be collected. The system needs to support quick review and navigation of large documents (in some cases more than 200 pages). The system needs to be available online to allow for easy review and collaboration. It needs to support mark-up and highlights that can be easily bookmarked. Lastly, it needs to support data entry and export of the final lease abstract.

4 Related Work

While a number of tools for linguistic annotation are available, none of the existing tools were able to meet the main system requirements described above.

A number of stand-alone annotation applications have been made available to the NLP research community. For example, Callisto (Day et al., 2004), MMAX2 (Müller and Strube, 2006), Wordfreak (Morton and LaCivita, 2003) are stand-alone tools which support annotation schema definitions. The GATE NLP framework (Cunningham et al., 2002) integrates an annotation interface to the framework’s Java GUI. A couple of annotation plug-ins have been developed for the Protégé Java framework - iAnnotateTab (Plimmer et al., 2010) and Knowtator (Ogren, 2006). Web-based annotation tools have also been developed to avoid the hassle of stand-alone applications and to streamline collaborative annotation efforts. For example, A.nnotate\(^3\), Bratt(Stenetorp et al., 2012), Marky(Perez-Perez et al., 2014), WebAnno\(^4\), FoLiA\(^5\),Djangology(Apostolova et al., 2010).

5 System Description

The system workflow and functionality are described in Figures 2 through 11.

---

\(^1\)Full automation of lease abstraction is not deemed feasible due to the complexity and associated liability of the task.

\(^2\)We utilize ABBY OCR technologies\(^\circ\).

\(^3\)http://a.nnotate.com/

\(^4\)https://code.google.com/p/webanno/

\(^5\)https://github.com/proycon/flat/
Figure 2: A secure online interface allows a user to log in and view a list of assigned documents. The documents were pre-processed through a 3rd party OCR engine. The interface supports full-text search and filtering based on various document attributes: status, submission and last modified timestamps.

Figure 3: Once a document is selected, the user is able to see the original document scan (images). The text version of the document is displayed as an invisible overlay matching the boundaries of the text in the underlying image. All document formatting, tables, markup, signatures, etc. are preserved. The dropdown shows a configurable list of color-coded abstraction fields.

Figure 4: New annotations are created by the user highlighting relevant document text and selecting the appropriate field (or associated short-cut). All operations are saved real-time on the server and explicit Save is not necessary. Overlapping annotations are supported as shown.

Figure 5: Existing annotations can be selected for editing by clicking on the annotated text in the left panel, or selecting the appropriate annotation in the navigation panel on the right. The user can use the navigation panel to sort annotations in chronological order (based on the submission timestamp of the annotation), or document order (based on the location of the annotation within the document).

Figure 6: Annotations can be deleted by selecting the X button in the navigation panel.

Figure 7: The abstract view of the navigation panel shows annotations grouped by abstract field types. For example, the abstractor can quickly navigate through all 9 sections relevant to the Landlord Operating Expenses abstract field and summarize the final abstract value in the corresponding text box.

Figure 8: The final abstract value can also be pasted directly from the text of the selected annotation using the paste button on the right. In this example, the abstractor can trim the text for Holdover rate to the desired value (150%) before confirming the final abstract value.
6 Conclusion

Supervised NER and text classification tasks could benefit expensive and labour-intensive legal document reviews tasks. A major obstacle in applying automation via ML to these tasks is the associated high cost of obtaining training data. At the same time, legal document abstraction tasks are by their nature a form of linguistic annotation and are performed on a large-scale across businesses. We presented a tool that supports efficient legal document abstraction following established workflows. At the same time, the tool covertly collects annotated textual data. The collected training data can then in turn be used to semi-automate the abstraction process.

References


