DeeperDive: The Unreasonable Effectiveness of Weak Supervision in Document Understanding

A Case Study in Collaboration with UiPath Inc.

Emad Elwany
Lexion
Seattle, USA
emad@lexion.ai

Allison Hegel
Lexion
Seattle, USA
allison@lexion.ai

Marina Shah
Lexion
Seattle, USA
marina@lexion.ai

Brendan Roof
Lexion
Seattle, USA
brendan@lexion.ai

Genevieve Peaslee
Lexion
Seattle, USA
genevieve@lexion.ai

Quentin Rivet
Lexion
Seattle, USA
quentin@lexion.ai

ABSTRACT

Weak supervision has been applied to various Natural Language Understanding tasks in recent years. Due to technical challenges with scaling weak supervision to work on long-form documents, spanning up to hundreds of pages, applications in the document understanding space have been limited. At Lexion, we built a weak supervision-based system tailored for long-form (10-200 pages long) PDF documents. We use this platform for building dozens of language understanding models and have applied it successfully to various domains, from commercial agreements to corporate formation documents.

In this paper, we demonstrate the effectiveness of supervised learning with weak supervision in a situation with limited time, workforce, and training data. We built 8 high quality machine learning models in the span of one week, with the help of a small team of just 3 annotators working with a dataset of under 300 documents. We share some details about our overall architecture, how we utilize weak supervision, and what results we are able to achieve. We also include the dataset for researchers who would like to experiment with alternate approaches or refine ours.

Furthermore, we shed some light on the additional complexities that arise when working with poorly scanned long-form documents in PDF format, and some of the techniques that help us achieve state-of-the-art performance on such data.

CCS CONCEPTS

• **Computing methodologies → Natural language processing**.

KEYWORDS

weak supervision, document understanding

1 BACKGROUND

At Lexion, we are often approached by partners who rely on our Document Intelligence expertise to solve difficult document understanding problems. UiPath, a leading enterprise automation software company, approached us in February 2022 with an exploratory project to evaluate the effectiveness of our platform. UiPath tasked us with extracting 8 key concepts from a set of 257 legal documents. These documents are “Collective Bargaining Agreements” that govern the relationship between labor unions and employers.

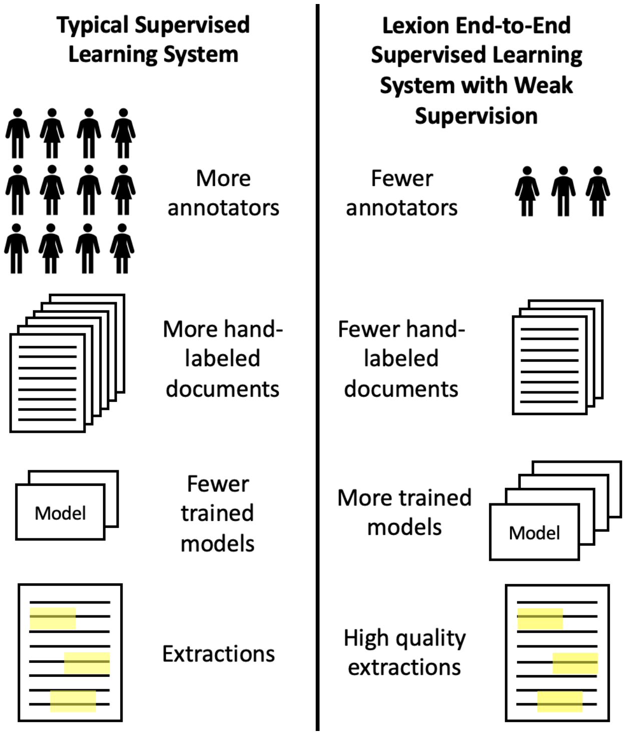
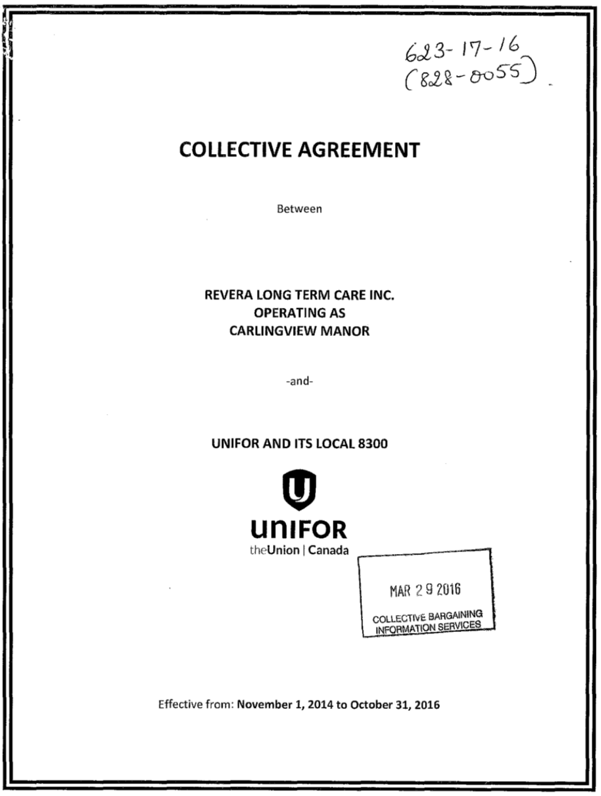

Typical Supervised Learning System
Lexion End-to-End Supervised Learning System with Weak Supervision
More annotators
Fewer annotators
More hand-labeled documents
Fewer hand-labeled documents
Model
Fewer trained models
More trained models
Model
Extractions
High quality extractions

Figure 1: We develop an end-to-end system using weak supervision and deep neural networks to extract information from legal contracts with accuracy that rivals typical supervised learning systems.

The goal was to see if we could build models to extract these 8 key concepts at high precision and recall with only one week of development. In addition to the time constraint, the dataset itself was particularly challenging. The concepts appear throughout the documents in an unstructured manner, and these documents were 50 to 100 pages long on average. They were poorly scanned PDF documents with no native text information. The target values can show up in tables or in prose. They are regularly spread out over paragraphs and are often interrupted by page boundaries, headers, footers and other elements, requiring a deep semantic understanding of the language to extract the correct values. We demonstrate that it is possible to achieve high precision and recall on this task despite constraints on time, workforce, and training data using an end-to-end supervised learning system that leverages weak supervision and deep neural models (Figure 1).

2 DATASET

Collective Bargaining Agreements are written contracts between an employer and a union representing the employees. The data used for this project consisted of 257 documents from the last 2 decades. Most of the documents were poorly scanned, and had complex layouts, obstructed text, and handwriting. Figure 2 and Figure 3 show some representative examples of the data.



623-17-16
(828-0055)

COLLECTIVE AGREEMENT

Between

REVERA LONG TERM CARE INC.
OPERATING AS
CARLINGVIEW MANOR

-and-

UNIFOR AND ITS LOCAL 8300

unifor
theUnion | Canada

MAR 29 2016
COLLECTIVE BARGAINING
INFORMATION SERVICES

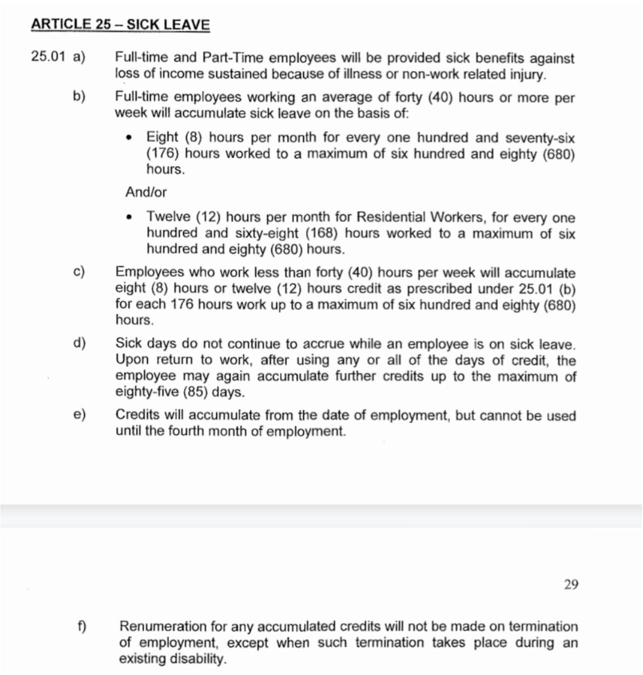
Effective from: **November 1, 2014 to October 31, 2016**

Figure 2: Cover page for a Collective Bargaining Agreement. Many documents in the dataset pose challenges such as poor scan quality, handwriting, tables, images, and stamps.

We extracted the following values from each agreement:

- **Employer Name:** The name of the employer/company that is party to the contract.
- **Union Name:** The name of the labor union that is party to the contract.
- **Agreement Start Date:** The beginning of the fixed term for the contract.
- **Agreement End Date:** The end of the fixed term for the contract.
- **Sick Leave Clause:** The entire section and/or subsection discussing sick leave.
- **Sick Leave Amount per Employment Status:** The number of hours, shifts, days, or other unit of sick leave provided to employees.
- **Sick Leave Unit per Employment Status:** The unit listed after amount of sick leave, which may be per some amount of time worked (e.g. 8 hours per 2 weeks worked).
- **Employment Status:** The type of employee being referred to (e.g. part-time, full-time, or all employees).

Upon receiving the data, we divided it into a train set (80% of the data), a development set (10%), and a test set (10%). We then ingested the data into our Document Intelligence platform. The development and test sets were manually annotated with perfect precision by our annotation team. Each document was first labeled by an annotator and then reviewed by a second annotator. The train set was annotated purely using weak supervision. Since long documents, particularly legal documents, are a particular challenge for natural language processing [2], we have published our dataset for future research.[1]



ARTICLE 25 – SICK LEAVE

25.01 a) Full-time and Part-Time employees will be provided sick benefits against loss of income sustained because of illness or non-work related injury.

b) Full-time employees working an average of forty (40) hours or more per week will accumulate sick leave on the basis of:

- Eight (8) hours per month for every one hundred and seventy-six (176) hours worked to a maximum of six hundred and eighty (680) hours.

And/or

- Twelve (12) hours per month for Residential Workers, for every one hundred and sixty-eight (168) hours worked to a maximum of six hundred and eighty (680) hours.

c) Employees who work less than forty (40) hours per week will accumulate eight (8) hours or twelve (12) hours credit as prescribed under 25.01 (b) for each 176 hours work up to a maximum of six hundred and eighty (680) hours.

d) Sick days do not continue to accrue while an employee is on sick leave. Upon return to work, after using any or all of the days of credit, the employee may again accumulate further credits up to the maximum of eighty-five (85) days.

e) Credits will accumulate from the date of employment, but cannot be used until the fourth month of employment.

29

f) Renumeration for any accumulated credits will not be made on termination of employment, except when such termination takes place during an existing disability.

Figure 3: An example of a Sick Leave Clause from a Collective Bargaining Agreement in the dataset. These agreements pose challenges including section and subsection hierarchy, headers and footers, and page breaks.

3 APPROACH

3.1 System Overview

Lexion's Document Understanding pipeline is comprised of multiple stages outlined in Figure 4. For this exercise, we leveraged many of

[1] https://drive.google.com/drive/u/0/folders/1rhglEd_IedBTJAF9G1KwdO2jii55FoeZ

our existing pre-trained models for the preliminary stages of the pipeline such as splitting PDFs that contain multiple agreements and detecting clauses. We focused primarily on training new models for the entity extraction and classification steps of the pipeline.

3.2 Weak Supervision

Our platform allows writing labeling functions using a domain-specific language. For an example of the syntax used by these labeling functions, see Table 1.

```
def label_sick_leave_hours(text) -> (start, end):
  for each sentence s in text:
    if s.starts("full time" or "part time") and
      s.contains("accumulate" or "accrue") and
      there exist tokens t1, t2 such that:
        if index(t1) - start(s) <= 5:
          index(t2) = index(t1) +1 and
          POS_TAG(t1) == "number" and
          NER_TAG(t2) == "time unit":
            return offsets(t1)
```

Table 1: An example of a labeling function. The domain-specific language allows annotators to create functions that are highly specific and yet flexible enough to achieve high recall with only a limited number of functions.

By writing between 10 and 20 labeling functions for each concept that needed to be extracted, the annotation team was able to achieve 87-100% annotation coverage for the training set in only a few days, with most models between 97-99% coverage. We then used the platform to train the target models, which produces both train set and development set metrics. The trained models use neural network entity extraction and classification models that utilize large transformer language models with task-specific layers on top [1, 8].

The annotation was completed by a team of 3 annotators over the span of 5 days. This time included not only writing labeling functions and annotating the test and development sets, but also doing schema and data discovery to understand the domain and gain the knowledge that was distilled into labeling functions.

3.3 Training Platform

The training platform allows us to train the full pipeline or specific nodes very quickly by picking which nodes of the pipeline to train, and specifying the configurations we'd like, including model architecture and hyperparameters (Figure 5).

In addition, the platform has metrics and deployment machinery built in (Figure 6), which allows us to quickly review the performance of individual models and roll them out to our user-facing Lexion interface.

3.4 Lexion Interface

We have found that a powerful user interface is an instrumental part of a document intelligence system. For that purpose, we have developed the Lexion interface (Figure 7) which makes it easy for non-technical users to view the results of document understanding extractions.

This interface doesn't just surface extractions to the end user, but also highlights the relevant language to provide explainability and confidence. It also allows users to verify the accuracy of extractions and correct mistakes, which is a powerful tool for closing the feedback loop between the user and the models. The user feedback we collect also allows further fine-tuning of the models.

4 RESULTS

With such a small and diverse dataset, by applying weak supervision for rapid annotation and a powerful neural network architecture, we were able to achieve impressive accuracy on the test set. We only evaluated the test set after training all the models, ensuring that it serves as a true blind set and was not subject to overfitting. Table 2 demonstrates the results we were able to achieve on each desired concept.

Concept	Dev P	Dev R	Dev F1	Test P	Test R	Test F1
Employer Name	88.9	79.0	83.7	93.0	81.0	86.6
Union Name	94.1	91.0	92.5	96.8	80.1	87.7
Start Date	98.9	94.8	96.8	93.0	95.5	94.2
End Date	93.0	98.8	95.8	91.6	96.4	94.0
Sick Leave Clause	74.0	78.0	76.0	85.0	73.0	78.0
Sick Leave Amount	97.5	65.7	78.0	90.3	79.3	84.4
Sick Leave Unit	89.6	71.5	79.4	81.0	77.1	79.0
Employment Status	89.1	62.9	73.3	78.1	70.3	73.8

Table 2: Precision, recall, and F1 score results on development and test data.

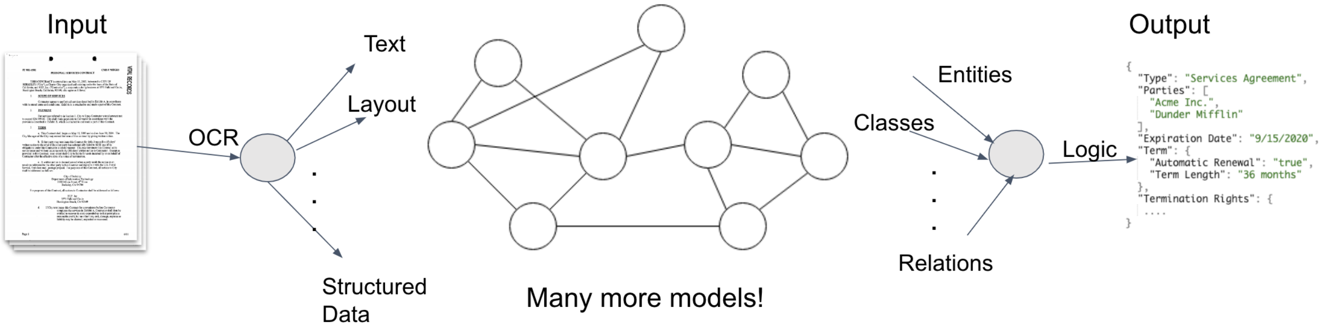
Our models performed best on the two date-based concepts, Start Date and End Date, likely because of the strong named entity support of the domain-specific language we use to create labeling functions. However, Employment Status was particularly challenging for the model because the contracts often discussed multiple statuses (e.g. both full-time and part-time) in the same clause.

The core differentiation of our system is the speed of model development. When the same techniques demonstrated in this paper are applied to larger datasets, for example the datasets that we use to train document understanding models on commercial agreements, the accuracy increases rapidly and often reaches F1 scores in the 0.85-0.95 range. On these large datasets (on the order of hundreds of thousands of documents), weak supervision is even more impactful since it would be prohibitively expensive to scale up model building with large amounts of annotation work.

5 SCALABLE WEAK SUPERVISION

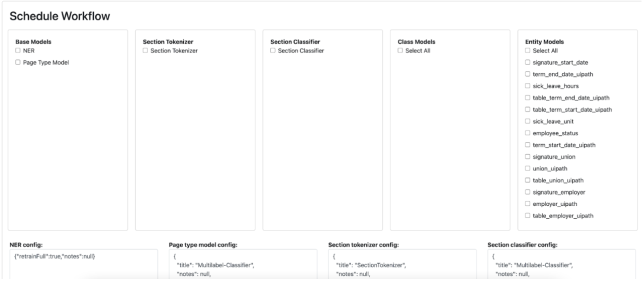
Typical supervised learning approaches require large amounts of labeled data. Weak supervision provides a method for generating labeled data at scale, while maintaining high accuracy. Most applications of weak supervision focus on shorter texts like question-answer pairs and sentences from scientific journal articles [4, 6, 7, 9]. One of the main challenges of applying weak supervision effectively to long-form documents like legal contracts is building a robust and interactive system that can quickly evaluate labeling functions on large volumes of documents [3].

Particularly for longer documents, incorporating metadata into labeling functions has been shown to improve performance [5]. Our



```
{
  "Type": "Services Agreement",
  "Parties": [
    "Acme Inc.",
    "Dunder Mifflin"
  ],
  "Expiration Date": "9/15/2020",
  "Term": {
    "Automatic Renewal": "true",
    "Term Length": "36 months"
  },
  "Termination Rights": {
   ....
}
```

Figure 4: The Lexion Document Understanding Pipeline.



Schedule Workflow

Base Models
- □ NER
- □ Page Type Model

Section Tokenizer
- □ Section Tokenizer

Section Classifier
- □ Section Classifier

Class Models
- □ Select All

Entity Models
- □ Select All
- □ signature_start_date
- □ term_end_date_uipath
- □ sick_leave_hours
- □ table_term_end_date_uipath
- □ table_term_start_date_uipath
- □ sick_leave_unit
- □ employee_status
- □ term_start_date_uipath
- □ signature_union
- □ union_uipath
- □ table_union_uipath
- □ signature_employer
- □ employer_uipath
- □ table_employer_uipath

NER config:
```
{"retrainFull":true,"notes":null}
```

Page type model config:
```
{
  "title": "Multilabel-Classifier",
  "notes": null,
```

Section tokenizer config:
```
{
  "title": "SectionTokenizer",
  "notes": null,
```

Section classifier config:
```
{
  "title": "Multilabel-Classifier",
  "notes": null,
```

Figure 5: Our training platform allows the team to rapidly train models individually or as an end-to-end pipeline, while specifying model architecture and hyperparameters.

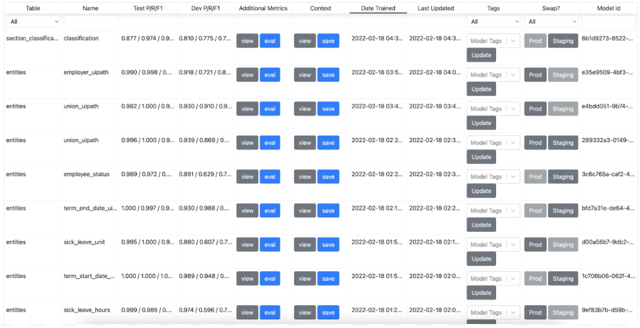
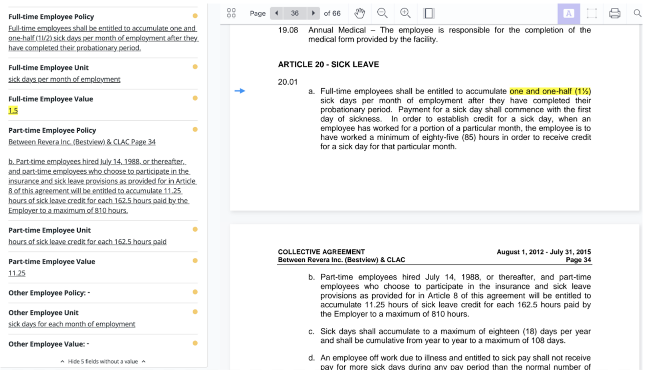


Table	Name	Test P/R/F1	Dev P/R/F1	Additional Metrics	Context	Date Trained	Last Updated	Tags	Swap?	Model Id
All								All	All	
section_classifica...	classification	0.877 / 0.974 / 0.9...	0.810 / 0.775 / 0.7...	view eval	view save	2022-02-18 04:3...	2022-02-18 04:3...	Model Tags Update	Prod Staging	6b1d9273-8522-...
entities	employer_uipath	0.990 / 0.998 / 0....	0.918 / 0.721 / 0.8...	view eval	view save	2022-02-18 03:5...	2022-02-18 04:0...	Model Tags Update	Prod Staging	e35e9509-4bf3-...
entities	union_uipath	0.982 / 1.000 / 0.9...	0.930 / 0.910 / 0.9...	view eval	view save	2022-02-18 03:4...	2022-02-18 03:4...	Model Tags Update	Prod Staging	e4bdd051-9b74-...
entities	union_uipath	0.996 / 1.000 / 0.9...	0.939 / 0.869 / 0....	view eval	view save	2022-02-18 02:2...	2022-02-18 02:3...	Model Tags Update	Prod Staging	289332a3-0149-...
entities	employee_status	0.989 / 0.972 / 0....	0.891 / 0.629 / 0.7...	view eval	view save	2022-02-18 02:2...	2022-02-18 02:3...	Model Tags Update	Prod Staging	3c6c765a-caf2-4...
entities	term_end_date_ui...	1.000 / 0.997 / 0.9...	0.930 / 0.988 / 0....	view eval	view save	2022-02-18 02:1...	2022-02-18 02:2...	Model Tags Update	Prod Staging	bfd7a31e-de64-4...
entities	sick_leave_unit	0.995 / 1.000 / 0.9...	0.880 / 0.607 / 0.7...	view eval	view save	2022-02-18 01:5...	2022-02-18 02:1...	Model Tags Update	Prod Staging	d00a56b7-9db2-...
entities	term_start_date_...	1.000 / 1.000 / 1.0...	0.989 / 0.948 / 0....	view eval	view save	2022-02-18 01:5...	2022-02-18 02:0...	Model Tags Update	Prod Staging	1c706b06-062f-4...
entities	sick_leave_hours	0.999 / 0.985 / 0....	0.974 / 0.596 / 0.7...	view eval	view save	2022-02-18 01:2...	2022-02-18 02:0...	Model Tags	Prod Staging	9ef83b7b-d59b-...

Figure 6: Our training platform offers detailed metrics on each model and allows the team to deploy newly-trained models with one click.

approach enriches documents with metadata that is used extensively in crafting labeling functions with high information density. In order to achieve this, we have made large investments into our data platform, as well as deviating from the status quo of using Python as the language for writing labeling functions in favor of more efficient domain-specific languages offered by open source frameworks. As a future goal, we hope to support labeling functions that have the full expressability of Python.



Full-time Employee Policy
Full-time employees shall be entitled to accumulate one and one-half (1I/2) sick days per month of employment after they have completed their probationary period.

Full-time Employee Unit
sick days per month of employment

Full-time Employee Value
1.5

Part-time Employee Policy
Between Revera Inc. (Bestview) & CLAC Page 34

b. Part-time employees hired July 14, 1988, or thereafter, and part-time employees who choose to participate in the insurance and sick leave provisions as provided for in Article 8 of this agreement will be entitled to accumulate 11.25 hours of sick leave credit for each 162.5 hours paid by the Employer to a maximum of 810 hours.

Part-time Employee Unit
hours of sick leave credit for each 162.5 hours paid

Part-time Employee Value
11.25

Other Employee Policy: -

Other Employee Unit
sick days for each month of employment

Other Employee Value: -

^ Hide 5 fields without a value ^

19.08 Annual Medical – The employee is responsible for the completion of the medical form provided by the facility.

ARTICLE 20 - SICK LEAVE

20.01

a. Full-time employees shall be entitled to accumulate one and one-half (1½) sick days per month of employment after they have completed their probationary period. Payment for a sick day shall commence with the first day of sickness. In order to establish credit for a sick day, when an employee has worked for a portion of a particular month, the employee is to have worked a minimum of eighty-five (85) hours in order to receive credit for a sick day for that particular month.

COLLECTIVE AGREEMENT — August 1, 2012 - July 31, 2015
Between Revera Inc. (Bestview) & CLAC — Page 34

b. Part-time employees hired July 14, 1988, or thereafter, and part-time employees who choose to participate in the insurance and sick leave provisions as provided for in Article 8 of this agreement will be entitled to accumulate 11.25 hours of sick leave credit for each 162.5 hours paid by the Employer to a maximum of 810 hours.

c. Sick days shall accumulate to a maximum of eighteen (18) days per year and shall be cumulative from year to year to a maximum of 108 days.

d. An employee off work due to illness and entitled to sick pay shall not receive pay for more sick days during any pay period than the normal number of

Figure 7: The Lexion interface displays the models' results in a user-friendly format, and allows users to correct any errors.

6 CONCLUSION

With the introduction of large language models, multimodal architectures that employ both natural language processing and computer vision as well as deep neural networks, there have been great advances in the accuracy of models that convert unstructured text into structured data. The cost and time of annotation remains one of the main obstacles to scaling document understanding. We demonstrate in this paper an end-to-end application of weak supervision that achieves high performance at scale despite requiring less time, training data, and annotation resources than typical supervised learning approaches to long document understanding tasks.

ACKNOWLEDGMENTS

This work wouldn't have been possible without the detailed problem specification and data provided by UiPath. We thank them for being great collaborators and for allowing us to publish the results and dataset publicly to further the research in this critical area of computer science.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[2] Emad Elwany, Dave Moore, and Gaurav Oberoi. 2019. BERT Goes to Law School: Quantifying the Competitive Advantage of Access to Large Legal Corpora in Contract Understanding. https://doi.org/10.48550/ARXIV.1911.00473

[3] Allison Hegel, Marina Shah, Genevieve Peaslee, Brendan Roof, and Emad Elwany. 2021. The Law of Large Documents: Understanding the Structure of Legal Contracts Using Visual Cues. (2021). https://doi.org/10.48550/ARXIV.2107.08128

[4] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 6086–6096. https://doi.org/10.18653/v1/P19-1612

[5] Dheeraj Mekala, Xinyang Zhang, and Jingbo Shang. 2020. META: Metadata-Empowered Weak Supervision for Text Classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 8351–8361. https://doi.org/10.18653/v1/2020.emnlp-main.670

[6] Alexander Ratner, Christopher De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data Programming: Creating Large Training Sets, Quickly. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) *(NeurIPS'16)*. Curran Associates Inc., Red Hook, NY, USA, 3574–3582.

[7] Jaeho Shin, Sen Wu, Feiran Wang, Christopher De Sa, Ce Zhang, and Christopher Ré. 2015. Incremental Knowledge Base Construction Using DeepDive. *Proc. VLDB Endow.* 8, 11 (jul 2015), 1310–1321. https://doi.org/10.14778/2809974.2809991

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[9] Zihan Wang, Dheeraj Mekala, and Jingbo Shang. 2021. X-Class: Text Classification with Extremely Weak Supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 3043–3053. https://doi.org/10.18653/v1/2021.naacl-main.242