A scalable approach to legal question answering

Zachary Bennett
LexisNexis
9443 Springboro Pike
Miamisburg OH 45342
zack.bennett@lexisnexis.com

Tony Russell-Rose
LexisNexis
30 Farringdon Street
London EC4A 4HH
tony.russellrose@lexisnexis.co.uk

Kate Farmer
LexisNexis
1801 Varsity Drive
Raleigh NC 27606
kathryn.farmer@lexisnexis.com

ABSTRACT

Lexis Answers is a question answering service deployed within a live production system. In this paper we provide an overview of the system, an insight into some of the key AI challenges, and a brief description of current evaluation techniques.

CCS CONCEPTS

• **Information systems ~Question answering** • **Information systems ~Search engine architectures and scalability** • *Information systems ~Information retrieval query processing*

KEYWORDS

Text mining, question answering, semantic retrieval.

1. INTRODUCTION

Legal researchers often satisfy their information needs through the use of keyword-based search applied to collections of natural language documents. Although much progress has been made in the optimization of relevance ranking functions (Liu, 2009), this approach relies on a commitment from users to examine individual documents in the results list and then identify and extract the required information. Consequently, there is a growing interest in developing systems that can more accurately interpret the user's search intent and deliver more targeted answers to support their information needs.

Lexis Answers is a new service that aims to provide answers to legal research questions. Instead of simply providing documents with potentially relevant sections highlighted, it extracts and delivers direct answers to legal questions in combination with a results list that is more precisely tuned to reflect the user's search intent. Lexis Answers incorporates machine learning (ML) and artificial intelligence (AI) technologies. The goal is to deliver a more effective legal research experience through improved analytics, visualization, and question answering (QA) capabilities.

In this paper we present a high level technical overview of the system, an insight into some of the key AI/ML challenges, and a brief description of current evaluation techniques and

performance data.

1.1 Legal question answering

Research into question answering has a long history dating back as far as the 1960s (Jurafsky and Martin, 2009), with more recent and notable milestones including the development of IBM Watson (Ferruci, 2011) and numerous consumer-oriented conversational agents, e.g. Siri, Cortana, Alexa, etc. However, with a few notable exceptions (e.g. Quaresma and Rodrigues, 2005), applications of such techniques to the domain of legal research are relatively rare. Moreover, fewer still have attempted to apply such functionality within the context of a live production system, with all the associated constraints of product alignment, architectural integration and quality assurance.

Lexis Answers has been designed from the outset to integrate directly with the Lexis Advance legal research platform. Queries are entered via a traditional search box which can accommodate keywords or terms and connectors (Boolean) searches. However, when a question-oriented intent is recognized, the system displays matching questions in the auto-suggest panel. If the user selects one of these (or enters a question of their own), a matching 'answer card' is displayed above the results list, which is also modified to reflect the user's search intent (see Figure 1).
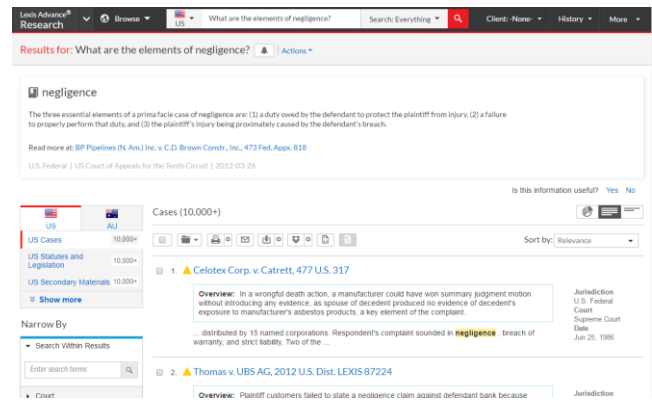

Figure 1. An answer card for the question 'what are the elements of negligence?'

In developing Lexis Answers we have initially focused on questions related to legal definitions and elements of legal concepts. The content for the answer cards is mined from our proprietary document corpora using a complex semi-automated pipeline (see below). The answers to some questions are dependent on jurisdiction.

Previous work has shown that although keyword-centric approaches may currently be dominant, many users demonstrate a willingness to adopt QA-style interactions and engage in conversational dialogs with systems that can learn from those interactions (Gupta & Gupta, 2012). Lexis Answers is currently deployed to a pilot user group who provide regular feedback via numerous channels such as focus groups, surveys and usage analytics.

2. SYSTEM OVERVIEW

Lexis Answers uses a large-scale Natural Language Processing (NLP) pipeline for extracting information from relevant sources. The pipeline ingests and parses documents, extracts metadata, identifies structural elements, and locates zones of the documents which may contain targeted information. The text of those zones is then converted by NLP tools into structured semantic and syntactic data, which is used in a hybrid rule-based and machine learning process to identify language that matches the targeted information need. When a potential answer is identified, the constituents of that answer are preserved as structured information, as well as the original text and source document information.

In the product, questions are suggested to the user based on matches between the user's query box input and the set of extracted answers. When a question is submitted, either via suggestion or direct query box input, a query analyzer classifies the intent of the question and identifies the query terms related to the intent. The intent and query terms are federated to services that execute structured searches to locate an appropriate response. A ranking algorithm selects the best answer based on ranking criteria and query interpretation confidence. The ranking criteria takes many aspects of legal understanding under consideration, including user profile and behavior information, document metadata, and candidate answer characteristics. The highest ranking response is displayed on the user's results page as the answer card.

3. AI CHALLENGES

Aside from common difficulties in processing legal text, the Lexis Answers NLP pipeline faces additional challenges raised by legal information extraction. First, the extraction must appropriately recognize contextual treatment that can render information irrelevant, such as indications of reported speech. Second, it must successfully identify context that is needed for disambiguation of the input. For instance, elements of "fraud" can be enumerated in different contexts, such as tort, contract, or criminal law, which might affect which answer is appropriate for a query. Lastly, the extraction process needs to generate additional data to support relevance ranking; for example, a statement of burden of proof for a claim might omit the responsible party, which might render the extraction less relevant for specific queries.

Because the bar for accuracy is high in a live, legal production system, additional information structures and a high degree of human involvement were then employed in the process. Data modelers and subject matter experts developed a content structure model to define the ways in which document structure drives interpretation of the text. Semantic types were defined for portions of documents to clarify the legal purpose of the text and therefore the appropriateness of extraction processes. And a legal information ontology was built to understand the relationships in the extracted information and elucidate what types of questions are "answerable". The relationships are critical for interpreting some of the extractions correctly; for example, a standard of review in an appellate court may be stated for a family of claims, but a user may pose a question regarding the standard of review for a specific claim. Lastly, taxonomies and controlled vocabularies for ontology instances, such as lists of criminal charges, were created by subject matter experts to aid in correctly labelling extracted information.

4. INITIAL EVALUATION

Due to the subjective nature of the task, the most indicative performance metric is a 4-point Likert-scale rating by expert annotators. To compute this metric, subject matter experts were given a test query and the answer returned by the system for that query. Three or more expert annotators then rated that answer for accuracy, relevance, and thoroughness. The final score was the average across all annotators for all query/answer pairs. Based on this method, the system is currently performing at 3.01/4.0.

The search results were also evaluated for relevance by subject matter experts. Three or more annotators were given the same result set in the context of the product and asked to assess relevance for a particular use case, such as case evaluation or finding a citation to include in a brief. The relevance scores were used to compute DCG (Discounted Cumulative Gain) for each test query. DCG improved 21% for the test query set with the implementation of Lexis Answers compared to the original baseline. We are satisfied with this result for an initial release, but hope to improve over time by incorporating feedback from real users.

5. CONCLUSIONS

Lexis Answers has proven to deliver a more relevant search experience for LexisNexis. In future work, we hope to increase the range of answer types and jurisdictions covered, to accommodate more complex scenarios such as questions with multiple concepts and relationships, and to develop more sophisticated techniques for search intent disambiguation and clarification.

REFERENCES

[1] Daniel Jurafsky and James H. Martin. 2009. Speech and Language Processing (2nd Edition). Prentice-Hall, Inc.

[2] David A. Ferrucci. 2011. IBM's Watson/DeepQA. SIGARCH Comput. Archit. News 39, 3 (June 2011), DOI= http://dx.doi.org/10.1145/2024723.2019525

[3] Tie-Yan Liu (2009), "Learning to Rank for Information Retrieval", Foundations and Trends in Information Retrieval, 3 (3): 225–331, doi:10.1561/1500000016

[4] Paulo Quaresma and Irene Pimenta Rodrigues. 2005. A Question Answer System for Legal Information Retrieval. In Proceedings of the 2005 conference on Legal Knowledge and Information Systems, IOS Press, Amsterdam, The Netherlands, The Netherlands, 91-100

[5] Gupta, Poonam, and Vishal Gupta. "A survey of text question answering techniques." International Journal of Computer Applications 53.4 (201)