Non-factoid Question Answering in the Legal Domain

Gayle McElvain*
gayle.mcevlain@capitalone.com
Capital One
McLean, Virginia, USA

George Sanchez, Don Teo, Tonya Custis
{first}.{last}@tr.com
Thomson Reuters
St. Paul, Minnesota, USA & Toronto, ON, Canada

ABSTRACT

Non-factoid question answering in the legal domain must provide legally correct, jurisdictionally relevant, and conversationally responsive answers to user-entered questions. We present work done on a QA system that is entirely based on IR and NLP, and does not rely on a structured knowledge base. Our system retrieves concise one-sentence answers for basic questions about the law. It is not restricted in scope to particular topics or jurisdictions. The corpus of potential answers contains approximately 22M documents classified to over 120K legal topics.

CCS CONCEPTS

• **Information systems → Question answering**; **Expert search**; *Query log analysis*; *Query reformulation*; • **Computing methodologies → Natural language processing**; **Artificial intelligence**; *Discourse, dialogue and pragmatics.*

KEYWORDS

question answering; legal question answering

ACM Reference Format:
Gayle McElvain and George Sanchez, Don Teo, Tonya Custis. 2019. Non-factoid Question Answering in the Legal Domain. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3331184.3331431

1 RELEVANCE TO SIRIP

IR in the Legal Domain is not well-covered in SIGIR. We present some of the challenges we have encountered in Legal IR such as jurisdictional coverage, the complexities of legal language, and the specifics of different legal practice areas, and how we have addressed them in industry.

Question Answering (QA) in the Legal domain requires a system that is precise and legally accurate, while also providing adequate recall across different jurisdictions (governing law may vary between jurisdictions). Such a system needs to be robust to the different information needs that arise across different legal specialties and practice areas, as well as to differences in language use between individual attorneys and the statutes of a particular jurisdiction.

Attorneys are continually trying to optimize their research time, trying to provide value for their clients by doing the most comprehensive research possible in the allowed number of billable hours. We present a Question Answering system for legal research that allows attorneys to zero in on the most salient points of law, related case law, and statutory law appropriate to their jurisdiction.

Since its launch in July 2018, 40% of Westlaw users have triggered the WestSearch Plus feature. When answers are presented to the user, there is a 52% clickthrough rate to see the full case, statute, or more answers related to their question.

2 SYSTEM OVERVIEW

As is typical, our QA system was trained on a large corpus of question-answer pairs. In total, we trained the production system on approximately 200K QA pairs, with each answer rated on a four point score (A, C, D, F). An average of three judgments were collected for each answer, and all judgments were supplied by attorney-editors with domain expertise.

Our system aims to provide conversationally fluent, concise one sentence answers for basic questions about the law. It is not restricted in scope to particular topics or jurisdictions. The corpus of potential answers contains about 22M documents classified to over 120K topics.

Logically, the QA system has three main components: Linguistic Analysis of Questions & Answers, Query Generation & Federated Search, and Question-Answer Pair Scoring.

2.1 Linguistic Analysis of Questions & Answers

Linguistic analysis is done on questions and answers to infer basic linguistic structure. This involves machine learning algorithms trained to predict parts of speech, noun and verb phrases, syntactic dependency relations, and semantic roles. The QA system employs open source models,[1] [2] trained on annotated sentences from a variety of sources not restricted to the legal domain [3].

To detect named entities and legal concepts in both questions and answers, we use a combination of gazetteer lookup taggers and statistical taggers trained with Conditional Random Fields.[3]

We classify a question's semantic intent to a set of predetermined semantic frames. A semantic frame is a coherent structure of related concepts, where the relationships between concepts tend to be realized according to prototypical patterns. This notion of frame is borrowed from Frame Semantics [4] and related notions in theories of Construction Grammar [5]. The process for identifying semantic frames was informed by editorial guidelines used to author the

*Work was done while employed at Thomson Reuters.

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331431

[1] https://opennlp.apache.org/
[2] https://emorynlp.github.io/nlp4j/
[3] http://www.chokkan.org/software/crfsuite

one-sentence summaries comprising the answer corpus. The frame of a question and its ideal answer should always be one in the same.

At runtime, questions are classified to a particular frame (or as "out-of-frame"). A Neural Network question classifier is trained on labeled (question, frame) pairs for each frame [1].

2.2 Query Generation & Federated Search

Search is a critical component in our QA system, functioning as the primary means by which to narrow the universe of potential answers for scoring. Linguistic analysis and feature generation for all questions against 22M potential answers is computationally expensive relative to the scoring methods used for search.

While no search strategy can precisely identify all possible answers to a question, the likelihood of retrieving a correct answer is increased by running multiple searches against different search engines. We execute three types of queries for each question against different search indices in two different search engines: 1) Natural language searches, derived from the question text; 2) Structured semantic searches, derived from the text, entities, and semantic frame information of the question; 3) More-like-this relevance feedback searches, derived from highly-ranked candidate answers.

A default Natural language search strategy is applied to all incoming questions. This type of search is run against answer indices created in both a proprietary search engine and Elasticsearch.

The question answering system leverages semantic search strategies for questions belonging to known frames. The questions are classified at runtime, but candidate answers can be classified offline and stored in a separate index. This enables search to target the collection of answers sharing the same semantic frame as the question. Depending on the frame of the question, multiple queries may be generated in order to target specific frame elements. Recognized entities and legal concepts in the question replace placeholders in frame-specific template queries to produce fully formed queries for execution against a search engine.

Finally, More-like-this search is used to widen the pool of potential answers after an initial set of candidates have been scored. This relevance feedback strategy is used mainly to expand coverage for specific jurisdictions. It involves searching for answers that closely match high scoring answers from outside the user's jurisdiction.

Document vectors constructed over the answer corpus are used to measure the semantic similarity between top-ranked answer candidates and more-like-this answer candidates. The approach used to generate document embeddings is based on the Paragraph Vectors model, also known as doc2vec, proposed by [7]. Models were trained on the answer corpus using an implementation provided by [6].

2.3 Question-Answer Pair Features & Scoring

Positional features capture the intuition that concepts in the question are likely to occur more closely together in correct answers than in incorrect answers. Distance is measured over the syntactic parse tree as well as over token and character offsets.

Answers that read like a natural answer to the question will typically put concepts from the question in English "topic" position near the beginning of the sentence. Highly-rated answers have a strong tendency to exhibit this pattern. Correct answers also often have question concepts near the root of the answer's syntactic parse tree. Both these tendencies are captured with topicality features.

All questions and answers are classified to a legal taxonomy with over 120K fine-grained categories. The classification scheme is quite complex, so user questions are generally underspecified relative to the taxonomy and correct answers can span multiple categories. In the same search result, however, there is a tendency for correct answers to have fewer distinct category classifications among them than incorrect answers. This association between question intent and taxonomic classification is leveraged by the system.

Feature scoring functions for the syntactic analyses of a question answer pair primarily measure overlap and alignment between the question and the answer sentence. Different features compute the alignment between noun phrases, dependency relations, and verb phrases. Various word embedding models trained on both open domain corpora and legal corpora are employed to measure semantic similarity within these structures.

All of the above features are combined in an ensemble model of weak learners [2]. This supervised model learns by example from labeled question answer pairs. At runtime, each QA pair is considered independently by the model and produces a score that represents the probability of that candidate being a correct answer for that question.

The last stage of the system determines whether or not to show an answer based on its probability score. Determining probability score thresholds is a business decision that weighs the relative cost of showing some incorrect answers against the cost of showing customers fewer answers (i.e., answering fewer customer questions).

Thresholds were set using 10-fold cross validation on all graded data. Thresholds were chosen by the business to: 1) maximize *Answered at*[4] metrics for correct answers (90% *Answered at* 3), 2) minimize *Answered at* metrics for **F** answers (1.5% *Answered at* 3), while 3) also balancing the system's coverage (the number of user questions for which answers are shown).

REFERENCES

[1] Piotr Bojanowski Armand Joulin, Edouard Grave and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. EACL, 427–431.

[2] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD, 785–794.

[3] Jinho D. Choi. 2016. Dynamic Feature Induction: The Last Gist to the State-of-the-Art. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. ACL, 271–281.

[4] Charles J. Fillmore. 2006. Frame semantics. Cognitive linguistics: Basic readings. In *Cognitive linguistics: Basic readings*, Dirk Geeraerts (Ed.). Walter de Gruyter, Berin, 34, 373–400.

[5] Adele E. Goldberg. 2006. In *Constructions at work: The nature of generalization in language*. Oxford University Pressr.

[6] Jey Han Lau and Timothy Baldwin. 2016. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. 78–86.

[7] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*. ICML, 1188–1196.

[4] Answered at is defined for each question's answer set such that it is the percentage of questions for which there is at least one of a particular label returned by the system at or above the rank indicated (so, at rank 1, *Answered at* 1 for **A**s is equivalent to Precision at 1).