Legal Knowledge and Information Systems
M. Araszkiewicz and V. Rodríguez-Doncel (Eds.)

doi:10.3233/FAIA190302

Improving the Processing of Question Answer Based Legal Documents

Saurabh CHAKRAVARTY [1], Maanav MEHROTRA,
Raja Venkata Satya Phanindra CHAVA, Han LIU, Matthew KRIVANSKY and
Edward A. FOX
Virginia Tech, Blacksburg, VA 24061 USA

Abstract.
In the legal domain, documents of various types are created in connection with a case. Some are transcripts prepared by court reporters, based on notes taken during the proceedings of a trial or deposition. For example, deposition transcripts capture the conversations between attorneys and deponents. These documents are mostly in the form of question-answer (QA) pairs. Summarizing the information contained in these documents is a challenge for attorneys and paralegals because of their length and form. Having automated methods to convert a QA pair into a canonical form could aid with the extraction of insights from depositions. These insights could be in the form of a short summary, a list of key facts, a set of answers to specific questions, or a similar result from text processing of these documents. In this paper, we describe methods using NLP and Deep Learning techniques to transform such QA pairs into a canonical form. The resulting transformed documents can be used for summarization and other downstream tasks.

Keywords. NLP, QA Normalization, Chunking, Deep learning, Legal Deposition

1. Introduction

Documents such as legal depositions comprise conversations between a set of two or more people, with the goal of identifying observations and the facts of a case. These conversations are in the form of discrete question-answer (QA) pairs. Like other general conversations, these documents are noisy, only loosely following grammatical rules.

Humans, because of their prior learning and experience, readily understand such documents since the number of types of questions and answers is limited. These types provide strong semantic clues that aid comprehension. Accordingly, we seek to leverage the QA types found, to aid textual analysis.

Classifying each QA pair type can ease the processing of the text, which in turn can facilitate downstream tasks like question answering, information retrieval, summarization, and knowledge graph generation. This is because special rules can be applied to each QA type, allowing transformations that are oriented to supporting existing NLP tools. This can facilitate text parsing techniques like constituency, syntax, and depen-

[1]Corresponding Author: Saurabh Chakravarty, Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, USA; Email:saurabc@vt.edu

dency parsing, and also enable us to break the text into different chunks based on part-of-speech (POS) tags using techniques like Chunking and Chinking.

Dialog Acts (DAs) [1,2] can represent the communicative intention behind a speaker's utterance in a conversation. Identifying the type of DA of each question and answer in a conversation [3] thus is a key first step in automatically determining intent and meaning.

Unfortunately, automatically transforming sentences based on DAs isn't straightforward. But, a possible solution is to transform the most prevalent combinations.

For a given type of QA pair, with its pair of types of question and answer DAs, we want to convert the QA pair into a canonical form. Table 1 shows an example question and answer, each with its respective dialog act, along with the desired canonical form.

Table 1. A QA pair with its canonical form.

Type	**Text**	**Dialog Act**
Question	Were you able to do physical exercises before the accident?	bin
Answer	Yes. I used to play tennis before. Now I cannot stand for more than 5 minutes.	y-d
Canonical Form	I was able to do physical exercises before the accident. I used to play tennis before. Now I cannot stand for more than 5 minutes.	-

As part of our work in Dialog Act (DA) classification [3], we observed common patterns associated with deposition QA pairs according to the different question and answer dialog acts. For each such common pattern, we can use traditional NLP parsing techniques like Chunking and Chinking [4] and create custom transformation rules to transform the text into a canonical form. Section 4.3.1 describes Chunking and Chinking in more detail. Section 4.3.2 describes an alternative approach for transformation into a canonical form, using Deep Learning.

The core contributions of this work are as follows.

1. An annotated dataset of QA pairs along with their Dialog Acts and canonical forms.
2. A collection of analysis and transformation methods using traditional NLP techniques like Chunking and Chinking.
3. A collection of Deep Learning based pre-trained sentence transformation models that can transform a QA pair into a canonical form.

2. Related Work[2]

Our earlier work [3] describes our ontology of Dialog Acts for legal depositions. This work also used two datasets to identify the various types of DA present in the deposition questions and answers. Deep Learning based classification methods were used to identify the DA associated with each of the question and the answer portion of a QA pair. For the current study, we re-purposed the DA classification methods in [3].

Once the types of DA for a QA pair have been identified, we want to transform the text into a canonical form. We have not been able to find any work that proposed a

[2]While we have completed an extensive literature review, limitations imposed on this submission have forced only mentioning a few of the many related works.

solution to this kind of problem. Traditional NLP based parsing techniques like Chunking and Chinking [4] can parse the constituents of a sentence based on the part-of-speech (POS) tags. These methods have been implemented in NLP libraries like NLTK [5] and spaCy [6] and have very good performance. Though the efficacy of these libraries is generally task based, an empirical analysis of the results helps make the best choice [7]. For our work we used the NLTK library for performing Chunking and Chinking.

Transforming a QA pair into a canonical form also can be formulated as a machine translation problem. Though we have the same source and target languages, the input and output differ in form. Works like [8] employ an encoder-decoder based approach to translate text from one language to other.

Work in COPYNET [9] added the idea of copying from the source input in sequence-to-sequence models. Pointer Generator Network (PGN) [10] is an abstractive summary generation system that used the same idea as COPYNET, but added more optimizations on how the summary is generated. It addressed two challenges: avoiding the generation of inaccurate text in the summaries, and controlling the repetition of text. During the training process, the system learns whether to generate or copy from the input sentence, and also to minimize the repetition while maximizing the probability of the generated sequence. We used the PGN architecture to transform a QA pair into a canonical form.

3. Datasets

For our work, we used DA combinations from datasets that each were a collection of depositions. We also curated the ground truth for our experiments for these datasets. The following sections describe these in more detail.

3.1. Dataset Description

We used depositions from a proprietary as well as a public dataset. The details for these datasets are as follows.

- **Mayfair Dataset** - This was a proprietary dataset that was provided to us by Mayfair Group LLC. This collection is comprised of 350 depositions. We randomly selected 10 depositions from this collection. Table 2 shows the distribution of the top 10 question-answer DA combinations across the Mayfair dataset.
- **Tobacco Dataset** - This dataset comes from the 14 million Truth Tobacco Industry Documents that are publicly accessible [11]. Over 2,000 of these are deposition transcripts. We randomly selected 8 depositions from this collection. Table 3 shows the distribution of the top 10 question-answer DA combinations across the tobacco dataset.

3.2. Dataset Annotation

One of the authors, along with volunteers selected by Mayfair, annotated the ground truth for the datasets. This involved annotating each QA pair with a simple sentence or other suitable canonical form of the QA pair. In our experiments we made use of about 4000 and 3300 annotated pairs for the Mayfair and tobacco datasets respectively.

Table 2. Distribution of the Top 10 DA combinations for the proprietary Mayfair dataset.

Question DA	Answer DA	# of samples	% of Total
wh	sno	517	13.00
bin	y	326	8.20
bin-d	y	322	8.10
bin	sno	277	6.96
bin	n	270	6.79
bin-d	sno	177	4.45
sno	sno	159	4.00
ack	sno	142	3.57
wh-d	sno	121	3.04
bin	y-d	99	2.49

Table 3. Distribution of the Top 10 DA combinations for the tobacco dataset [11].

Question DA	Answer DA	# of samples	% of Total
bin-d	sno	454	13.58
bin	sno	441	13.19
wh	sno	297	8.88
bin-d	y	235	7.02
bin	y	183	5.47
bin	n	143	4.27
sno	sno	143	4.27
bin	y-d	118	3.52
bin	dno	95	2.84
bin-d	y-d	92	2.75

4. Methods

4.1. Dialog Acts:

For our task of transformation, classifying the Dialog Acts (DAs) [1,2] would aid in isolating and grouping QA pairs of similar type. Custom rules can be developed for each DA type to process a conversation QA pair and transform it into a suitable form for subsequent analysis. Using methods to classify the DAs in a conversation thus would help us delegate the transformation task to the right transformer method. We have used the ontology and the methods in [3] to classify the DAs in our dataset.

4.2. Pre-processing:

The text in the QA pairs contained noise which needed to be removed to perform the transformation step in an efficient way. Table 4 shows some sample questions with the noise that we needed to remove via pre-processing.

For some DAs, the question and answer text also consisted of a well formed sentence in the beginning and the end, as shown in Table 5. We used text-processing techniques along with regular expression based rules to separate the declarative part from the question and the answer.

Table 4. Questions, with the noisy text in bold.

You also mentioned earlier that he busted his lips**; is that correct?**
Okay. So you mentioned you had a son**; correct?**
I see. So, did you think it was the bartender?

Table 5. Questions and answers that include a well formed sentence. Declarative parts shown in bold.

Text	Dialog Act
And the damage that you showed earlier in the diagram, you said that damage was accidental?	bin-d
And a fracture that runs through the whole arm joint is a pretty severe fracture. When was the examination done?	wh-d
Yes. **We sent out this to that operating company.**	y-d
No. **I did not read any depositions or I think the second part is kind of general, but I haven't read any depositions.**	n-d

4.3. Transformation

We used two different methods to transform the QA pair into a canonical form. The following sections describe the methods in more details.

4.3.1. Transformation via Chunking and Chinking

Chunking refers to the process of extracting chunks from a sentence based on certain POS tag rules. These rules are represented using simple regular expressions. Chinking refers to the process of defining what is not to be included in a chunk. A Chunking process creates chunks and Chinking breaks up those chunks into more granular parts using rules. Referring to the example present in Table 1, we started with the question text and created a simple sentence parse tree as shown in Figure 1. Then we broke it up into a chunk based on a preposition rule of *"<.*>?<PRP ><.*>?."* This rule specifies that any preposition that has any POS tag before and after it should be extracted as a chunk. In this case, it extracted "Were" and "able" that were before and after the preposition word. Figure 2 shows the chunk formed as part of the Chunking process.

Figure 1. Sentence root.

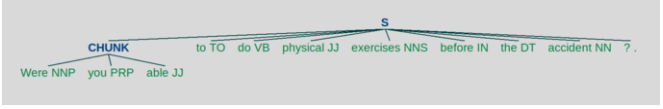

Figure 2. Extracting a chunk based on a rule.

For transformation to a canonical form, we needed to transform the identified chunk into a first person description. This description will be from the perspective of the depo-

nent. The transformed sentence in this case would be *"I was able to do physical exercises before the accident"*. We swapped the position of "were" and "you" in the chunk tree and transformed "you" to "I" and "were" to "was". For each of these simple transformations of a QA pair word to a canonical form word, we created a dictionary entry to keep track of that transformation. The dictionary was expanded to account for different transformations that were required for other words that needed to be transformed. We iteratively improved our transformation based on the results we observed from the data. We developed specific methods for each combination of a question and answer DA.

4.3.2. Transformation via Deep Learning.

The Deep Learning based transformation was implemented with a prototype we devised to evaluate the feasibility of using Deep Learning based methods. There are no known works that have addressed our exact problem, so we investigated how Deep Learning based models would perform for this task. We used the OpenNMT Toolkit [12] to train sentence transformers for the different combinations of DA.

Deep Learning models are dependent on a large number of training examples; this is more pronounced for sequence-to-sequence models where there are a large number of parameters in play. Since the amount of training data we could obtain was limited, we focused our collection of training data on a particular set of the combinations of DA. In particular, we only developed Deep Learning based methods for the combinations of [bin, y], [bin, n], [bin, y-d], and [bin, n-d].

4.4. Evaluation Methods.

Evaluation of text processing and transformation is much more difficult than for simple classification since the results are often subjective. For our preliminary evaluation studies, we started by using ROUGE-1/2 scores and sentence similarity for evaluation. Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [13] can be used to compare generated sentences with the canonical forms annotated by human actors. We used the ROUGE-1 and ROUGE-2 scores, which measure how much generated sentences overlap with the uni-gram and bi-gram representation of the annotated canonical forms.

Another evaluation metric we used is sentence similarity. Transforming a pair of sentences to their vector space representations and measuring their cosine-similarity can be used to measure sentence similarity. For that transformation, we used InferSent [14] to generate sentence embedding vectors; it is a based on fastText[15] word embeddings.

4.5. Experimentation

For our experiments, we considered the top 11 DA classes for the proprietary dataset as given in Table 2. The top 11 DA combinations represented more than 65% and 60% of the total data for the proprietary and tobacco datasets, respectively. This was a good set to target for our work. The DA combinations that we left out each represented less than 3% of the data. We plan to develop methods for these DA combinations in future work.

We developed transformation methods involving Chunking methods for 10 of the 11 DA classes. Regarding the ["bin-d", "sno"] DA combination, we found the question text to be problematic for transformation via our methods. The DAs for most of the questions were incorrectly classified as "bin-d", whereas it had a mix of "bin-d", "wh-d" and "sno".

For this reason, we omitted occurrences of the [“bin-d”, “sno”] DA combination from our experiments. We will address this omission in future work.

Table 6 describes the transformation methods used for our experiments.

Table 6. The transformation methods used for the experiments.

Method	Description
Just Answer	In this method we used the answer as is.
Question and Answer	In this method we used the combination of the concatenated question and answer text.
Chunking based Transformation	In this method we performed DA classification followed by Chunking based transformation.
Deep Learning based Transformation	In this method we performed DA classification followed by Deep Learning based transformation.

5. Results

5.1. Experiment Results and Analysis

The following discussion is of studies with the Mayfair dataset. Table 7 shows the results of the transformation experiments for the four different methods. We calculated the ROUGE-1(R-1)/2(R-2) and the similarity (Sim) scores between the ground truth and the generated sentence. We averaged the scores across all of the samples, for each DA combination. The following sections discuss the results in more detail for each method.

Table 7. Evaluation Results. Best results are highlighted in bold.

Qstn DA	Ans DA	Just Answer			Q+A			Chunking		
		R-1	R-2	Sim	R-1	R-2	Sim	R-1	R-2	Sim
wh	sno	0.73	0.67	0.79	0.77	0.66	**0.87**	**0.78**	**0.70**	0.86
bin	y	0.09	0.03	0.29	0.75	0.56	0.84	**0.85**	**0.70**	**0.90**
bin-d	y	0.016	0.002	0.11	0.81	0.70	0.90	**0.9**	**0.81**	**0.93**
bin	sno	0.67	0.63	0.76	0.83	0.75	0.90	**0.84**	**0.79**	**0.91**
bin	n	0.08	0.04	0.36	0.72	0.54	0.81	**0.83**	**0.70**	**0.91**
sno	sno	0.67	0.62	0.73	**0.9**	**0.85**	**0.95**	0.85	0.80	0.92
ack	sno	**0.98**	**0.98**	**0.99**	0.94	0.93	0.94	**0.98**	**0.98**	**0.99**
wh-d	sno	**0.82**	**0.78**	**0.87**	0.64	0.57	0.78	**0.82**	**0.78**	**0.87**
bin	y-d	0.55	0.47	0.68	**0.78**	**0.65**	**0.87**	0.75	**0.65**	0.82
bin	n-d	0.45	0.31	0.74	**0.59**	**0.43**	**0.80**	0.54	0.39	0.78

5.1.1. Use answer (results for top 5 combinations):

- *wh |sno* - The transformer performance was quite reasonable for both the ROUGE scores and similarity. For the best scores, we observed that the answer is descriptive and has a good overlap with the ground truth. For the worst scores, we observed that the answer was short and lacked the context that was present in the question
- *bin |y* - The transformer performance was poor for this case. This happens because the answer DA is “y” and in such cases the answer is in the form of “yes” or “yeah”, which does not contain enough context to match well with the ground truth.

- *bin-d |y* - The transformer performance was poor for this case, as with the previous one. The scores are also of similar nature and for the same reasons.
- *bin |sno* - The transformer performance was quite reasonable. The reasoning for this is similar to what was dicussed in wh |sno pair.
- *bin |n* - The transformer performance was poor for this case, similar to the bin |y combination. The scores are also of similar nature and for the same reasons.

5.1.2. Use question and answer (results for each DA combination):

- *wh |sno* - The transformer performance was very good for both the ROUGE scores and similarity. For the best scores, we observed that the answer is descriptive and has a high chance of having a good overlap with the ground truth. For the worst scores, we observed that the generated text contained the text from both question and the answer, whereas the ground truth was a good paraphrase of the same.
- *bin |y* - The transformer performance was very good for both the ROUGE scores and similarity. This happens because the answer DA is "y" and in such cases the answer is in the form of "yes" or "yeah", but the question contains enough context to have a good overlap with the ground truth.
- *bin-d |y* - The transformer performance was very good for this case, similar to the previous one. The scores are a little better, but for the same reason that the question and answer together in one sentence is bound to have good overlap and similarity with the ground truth.
- *bin |sno* - The transformer performance was very good for this case, similar to the previous one.
- *bin |n* - The transformer performance was reasonably good for this case. We observed higher scores for simple questions and long answer combinations. This because a combination of the two provides enough context. For the worst scores, we observed a high similarity score but a poor ROUGE-2 score. The generated sentence had a very poor bi-gram overlap with the ground truth.

5.1.3. Transformation via Chunking (results for each DA combination):

- *wh |sno* - The transformer performance was very good for both the ROUGE scores and similarity. For the other methods there were very rare or no occurrences of perfect ROUGE-2 scores. This underlines that the Chunking based methods had a good paraphrasing ability that matched the annotated ground truth. For the worst scores, we observed that the generated text was a good paraphrase of the question and answer, but it was not of the exact form as the ground truth.
- *bin |y* - The transformer performance was very good for this case, similar to the *wh |sno* case.
- *bin-d |y* - The transformer performance was very good for this case, similar to the previous one.
- *bin |sno* - The transformer performance was very good for this case. There were many instances of perfect ROUGE-2 scores. For the worst scores we observed that the Chunking based transformers were not able to break the QA pair using the predefined grammar rules and hence emitted the answers for these cases.
- *bin |n* - The transformer performance was reasonably good for this case. It was the best among all the methods used for the ROUGE scores and similarity. There were

many instances of perfect ROUGE-2 scores. For the worst scores, the Chunking based method had challenges with the grammar and some generated bi-grams had an incorrect form.

5.1.4. Transformation using Deep Learning:

We broke the dataset into a 70-20-10 proportion for training, validation, and test. Separate models were trained using the annotated data which was run for all 4 DA combinations. The results as shown in Table 8. The modest results could be attributed to the fact that we had very little training data to train with. The results do indicate a potential to improve with more training data. We plan to address this in our future work.

Table 8. Deep Learning results.

Question DA	Answer DA	ROUGE-1	ROUGE-2	Sentence Similarity
bin	y	0.6	0.38	0.73
bin	n	0.71	0.54	0.83
bin	y-d	0.48	0.26	0.74
bin	n-d	0.44	0.24	0.67

6. Conclusion and Future Work

We developed methods to transform a QA pair in a legal deposition to a canonical form. We used traditional NLP based techniques like Chunking and Chinking, along with methods based on Deep Learning. We found that the transformation methods based on Chunking had the best ROUGE-2 scores in 8 of the 10 DA combinations and had the best semantic similarity scores in 6 out of the 10 DA combinations. For most of the other comparisons, NLP techniques were competitive with the other best results.

To confirm the findings reported above for the Mayfair dataset, we ran additional experiments on the tobacco dataset. The results indicated equally good transformation performance in 8 of the 10 DA classes for the Chunking based methods. This indicates generality of the transformation methods across datasets.

As per our knowledge, this is the first work of its kind that transforms a QA pair into a canonical form. Given the encouraging results, we plan to improve it further and scale up the experiments with a larger corpora and additional evaluations.

We plan to improve the DA classification by adding a pre-processing step so that it can break a long question into a series of statements and questions. This would allow the classifier to be applied to shorter texts, which should result in increased DA accuracy.

We also plan to generate word embeddings for the legal domain, especially for depositions. We can use the BERT [16] system to train on a large deposition corpora and learn embeddings that are specific to legal depositions.

For the Deep Learning based transformers, we plan to train with more data and more DA combinations to improve transformation efficacy. Using grammatical correctness as a constraint for the generation of transformed text should improve results further.

We plan to refine our evaluation methods by using human actors to subjectively evaluate the quality of the transformed sentences using criteria like readability, context and polarity retention, and grammatical correctness.

Acknowledgments. This work was made possible by Virginia Tech's Digital Library Research Laboratory (DLRL). Data in the form of legal depositions was provided by Mayfair Group LLC, which also managed obtaining annotations. In accordance with Virginia Tech policies and procedures and our ethical obligations as researchers, we are reporting that Dr. Edward Fox has an equity interest in Mayfair Group LLC, whose data was used in this research. Dr. Fox has disclosed those interests fully to Virginia Tech, and has in place an approved plan for managing any potential conflicts arising from this relationship.

References

[1] D. Jurafsky, E. Shriberg, B. Fox and T. Curl, Lexical, prosodic, and syntactic cues for dialog acts, in: *Stede, Manfred, Leo Warner, and Eduard Hovy (eds.): Discourse Relations and Discourse Markers. Proceedings of the workshop, 15 August, Montreal, Quebec, Canada: COLING-ACL '98.*, New Brunswick, NJ: Association for Computational Linguistics, 1998, pp. 114–120.

[2] J. Williams, A belief tracking challenge task for spoken dialog systems, in: *NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012)*, 2012, pp. 23–24.

[3] S. Chakravarty, R.V.S.P. Chava and E.A. Fox, Dialog Acts Classification for Question-Answer Corpora, in: *Proceedings of the Third Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2019), June 21, 2019, Montreal, QC, Canada*, 2019.

[4] F. Zhai, S. Potdar, B. Xiang and B. Zhou, Neural models for sequence chunking, in: *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[5] E. Loper and S. Bird, NLTK: the natural language toolkit, *arXiv preprint cs/0205028* (2002).

[6] spaCyCommunity, *spaCy: Industrial-Strength Natural Language Processing in Python*, 2016, `https://spacy.io`.

[7] F.N.A. Al Omran and C. Treude, Choosing an NLP library for analyzing software documentation: a systematic literature review and a series of experiments, in: *Proceedings of the 14th International Conference on Mining Software Repositories*, IEEE Press, 2017, pp. 187–197.

[8] K. Cho, B. van Merrienboer, D. Bahdanau and Y. Bengio, On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, in: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014, pp. 103–111.

[9] J. Gu, Z. Lu, H. Li and V.O. Li, Incorporating copying mechanism in sequence-to-sequence learning, *arXiv preprint arXiv:1603.06393* (2016).

[10] A. See, P.J. Liu and C.D. Manning, Get to the point: Summarization with pointer-generator networks, *arXiv preprint arXiv:1704.04368* (2017).

[11] U. Library and Center for Knowledge Management, *Truth Tobacco Industry Documents*, 2002, `https://www.industrydocuments.ucsf.edu/tobacco`.

[12] G. Klein, Y. Kim, Y. Deng, J. Senellart and A.M. Rush, Opennmt: Open-source toolkit for neural machine translation, *arXiv preprint arXiv:1701.02810* (2017).

[13] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: *Text Summarization Branches Out*, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81.

[14] A. Conneau, D. Kiela, H. Schwenk, L. Barrault and A. Bordes, Supervised Learning of Universal Sentence Representations from Natural Language Inference Data, in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 670–680.

[15] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch and A. Joulin, Advances in Pre-Training Distributed Word Representations, in: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[16] J. Devlin, M. Chang, K. Lee and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *CoRR* **abs/1810.04805** (2018).