Dialog Acts Classification for Question-Answer Corpora

Saurabh Chakravarty
saurabc@vt.edu
Virginia Tech
Blacksburg, VA

Raja Venkata Satya Phanindra Chava
chrvsp96@vt.edu
Virginia Tech
Blacksburg, VA

Edward A. Fox
fox@vt.edu
Virginia Tech
Blacksburg, VA

ABSTRACT

Many documents are constituted by a sequence of question-answer (QA) pairs. Applying existing natural language processing (NLP) methods such as automatic summarization to such documents leads to poor results. Accordingly, we have developed classification methods based on dialog acts to facilitate subsequent application of NLP techniques. This paper describes the ontology of dialog acts we have devised through a case study of a corpus of legal depositions that are made of QA pairs, as well as our development of machine/deep learning classifiers to identify dialog acts in such corpora. We have adapted state-of-the-art text classification methods based on a convolutional neural network (CNN) and long short term memory (LSTM) to classify the questions and answers into their respective dialog acts. We have also used pre-trained BERT embeddings for one of our classifiers. Experimentation showed we could achieve an F1 score of 0.84 on dialog act classification involving 20 classes. Given such promising techniques to classify questions and answers into dialog acts, we plan to develop custom methods for each dialog act, to transform each QA pair into a form that would allow for the application of NLP or deep learning techniques for other downstream tasks, such as summarization.

1 INTRODUCTION

Documents such as legal depositions contain conversations between a set of two or more people, aimed at identifying observations and the facts of a case. The conversational actors are aware of the current context, so need not include important contextual clues during their communication. Further, because of that awareness, their conversations may exhibit frequent context shifts.

These conversations are in the form of rapid fire question-answer (QA) pairs. Like many conversations, these documents are noisy, only loosely following grammatical rules. Often, people don't speak using complete or well-formed sentences that can be comprehended in isolation. There are instances where a legal document is transcribed by a court reporter and the conversation contains words like "um" or "uh" that signify that the speaker is thinking. In many of the instances, there is an interruption that leads to incomplete sentences being captured or a sentence getting abandoned altogether.

These characteristics of QA conversations make it difficult to apply popular NLP processing methods, including co-reference resolution and summarization techniques. For example, there is the challenge of identifying key concepts using NLP based rules. In many corpora, the root words that are most prevalent in sentences help identify the core concepts present in a document. These core concepts help text processing systems capture information with high precision. However, traditional NLP techniques like syntax parsing or dependency trees sometimes struggle to find the root of conversational sentences because of their form.

Humans, on the other hand, readily understand such documents since the number of types of questions and answers is limited, and these types provide strong semantic clues that aid comprehension. Accordingly, we seek to leverage the types found, to aid textual analysis.

Defining and identifying each QA pair type would ease the processing of the text, which in turn would facilitate downstream tasks like question answering, summarization, information retrieval, and knowledge graph generation. This is because special rules could be applied to each type of question and answer, allowing conversion oriented to supporting existing NLP tools. This would facilitate text parsing techniques like constituency and dependency parsing and also enable us to break the text into different chunks based on part of speech (POS) tags.

Dialog Acts (DA) [19, 41] represent the communicative intention behind a speaker's utterance in a conversation. Identifying the DA of each speaker utterance in a conversation thus is a key first step in automatically determining intent and meaning. Specific rules can be developed for each DA type to process a conversation QA pair and transform it into a suitable form for subsequent analysis. Developing methods to classify the DAs in a conversation thus would help us delegate the transformation task to the right transformer method.

Text classification using deep learning techniques has rapidly improved in recent years. Deep neural network based architectures like Recurrent Neural Network (RNN) [13], Long Short Term Memory (LSTM) [17], and Convolutional Neural Network (CNN) [16] now outperform traditional machine learning based text classification systems. For example, LSTM and CNN networks help capture the semantic and syntactic context of a word. This enables the systems based on LSTM and CNN to model word sequences better. There have been various architectures in the area of text classification which use an encoder-decoder [8] based model for learning. Systems using CNNs [2, 7, 10, 22, 34] or LSTMs [7, 39] have had significant performance improvements over the previously established baselines in text classification tasks like sentiment classification, machine translation, information retrieval, and polarity detection. Accordingly, we focus on deep learning based text classification techniques and fine-tune them for our task of DA classification.

The core contributions of this paper are as follows.

In: Proceedings of the Third Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2019), June 21, 2019, Montreal, QC, Canada.

Published at http://ceur-ws.org

(1) A Dialog Act ontology that pertains to the conversations in the legal domain.
(2) An annotated dataset that will be available for the research community.
(3) Classification methods that use state-of-the-art techniques to classify Dialog Acts, and which have been fine-tuned for this specific task.

2 RELATED WORK

Early work on Dialog Act Classification [1, 14, 18, 23, 25, 28, 38, 40] used machine learning techniques such as Support Vector Machines (SVM), Deep Belief Network (DBN), Hidden Markov Model (HMM), and Conditional Random Field (CRF). They used features like speaker interaction and prosodic cues, as well as lexical, syntactic, and semantic features, for their models. Some of the works also included context features that were sourced from the previous sentences. Work in [36, 38] used HMM for modeling the dialog act probabilities with words as observations, where the context was defined using the probabilities of the previous utterance dialog acts. Work in [12, 18] used DBN for decoding the DA sequences and used both the generative and the conditional modeling approaches to label the dialog acts. Work in [6, 12, 21, 32] used CRF to label the sequences of dialog acts.

The sentences in the QA pairs need to be modeled into a vector representation so that we can use them as features for text classification. Availability of rich word embeddings like word2vec [30] and GloVe [31] have been effective in text classification tasks. These embeddings are learned from large text corpora like Google News or Wikipedia. They are generated by training a neural network on the text, where the objective is to maximize the probability of a word given its context, or vice-versa. This objective helps the neural network to group words that are similar in a high-dimensional vector space. Work based on averaging of the word vectors [5] in a sentence has given good performance in text classification.

In late 2018, Google developed BERT (Bidirectional Encoder Representations from Transformers) [11], a powerful method for sentence embeddings. It was pre-trained on a massive corpus of unlabeled data to build a neural network based language model. This allows BERT to achieve significantly higher performance for classification tasks which have a small task-specific data-set. The authors argued that the current deep learning based language models to generate embeddings are unidirectional and there are challenges when we need to model sentences. Tasks such as attention based question answering require the architecture to attend to tokens before and after, during the self-attention stage. The core contribution was the generation of pre-trained sentence embeddings that were learned using the left and right context of each token in the sentence. The authors also proposed that these pre-trained embeddings can be used to model any custom NLP task by adding a final fully connected neural network layer and modeling the network output to the task at hand. There is no need to create a complex network architecture. BERT internally uses the multi-layer network or “transformer” presented in [37] to model the input text and the output embedding. The transformer involves six layers of attention, followed by normalization and a feed-forward layer as an encoder, and the same layers plus an added masked attention layer for the decoder. The attention layer in the encoder and decoder builds self-attention on the input and output words, respectively, to learn what words are important. The masked attention layer in the decoder learns the attention only until the token in the output that has already been generated by the decoder so far. To train the model, the work involved learning on two tasks. The first task was to guess a masked word in a sentence, where each sentence was from a large corpus. The authors removed a word randomly from a sentence and trained the model to predict the right word. The second task was to predict the following sentence for a given sentence, from a choice of four sentences. The training was performed using the Google Books Corpus (with 800M words) [27] and English Wikipedia (with 2,500M words) [9]. The work obtained new state-of-the-art results on 11 NLP tasks as part of General Language Understanding Evaluation (GLUE), and was very competitive in other tasks.

Recent works like [20, 26, 33] use deep neural networks to classify the dialog acts. These works used models like CNN and LSTM to model the context for a sentence. Work in [20] used a CNN+LSTM model for the DA classification and slot-filling task using two different datasets. They obtained a negligible improvement for one of the datasets and a significant improvement for the other. Work in [33] used a recurrent CNN based model to classify the DAs, and obtained a 2.9% improvement over the LM-HMM baseline. Work in [26] used RNN and CNN based models for DA classification along with the DA labels of the previous utterances to achieve state-of-the-art results in the DA classification task.

3 METHODS

As part of our methods, we defined an ontology of dialog acts for the legal domain. Each sentence in the conversation was classified into one of the classes. The following sections describe the ontology and classification methods in more detail.

3.1 Dialog Act Ontology

After a thorough analysis of the conversation QA pairs in our dataset of depositions, two researchers refined a subset of the dialog acts found in [19]. These researchers also added additional dialog acts to our ontology for the questions and answers, again based on their analysis of the depositions. The following sections present more details.

3.1.1 Question specific dialog acts. Table 1 shows the different dialog acts that we have defined for the questions in the depositions.

We expanded the “wh” category, which covers many of the DAs in a deposition, into sub-categories. This would enable specific comprehension techniques to be used on each sub-category as the sentences are varied for each of the sub-categories. Table 2 lists and describes each sub-category for the “wh” parent category.

3.1.2 Answer specific dialog acts. Table 3 shows the different dialog acts that we have defined for the questions in the depositions.

3.2 Dialog Act Classification

We used different classifiers based on deep learning that have achieved state-of-the-art results in multiple other tasks. We also used simple classifiers that used sentence embeddings followed by

Category	Description	Example
wh	This is a wh-* kind of question. These questions generally start with question words like who, what, where, when, why, how, etc.	What time did you wake up on the morning the incident took place?
wh-d	This is also a wh-* kind of question. But if there is more than one statement in a what question, it is a what-declarative question. These questions have some information prior to the actual question which relates to the question.	You said generally wake up at 7:00 am in the morning. But what time did you wake up on the morning the incident took place?
bin	This is a binary question. These are questions that can be answered with a simple “yes” or “no”.	Is that where you live?
bin-d	This is a binary-declarative question which can also be answered with a “yes” or a “no”. But, in a binary-declarative question, the person who asks the question knows the answer but asks for verification. In contrast, a binary question indicates the examiner seeks to know which is the actual answer.	That is where you live, right?
qo	This is an open question. These questions are general questions which are not specific to any context. These questions are asked to know the opinions of the person who is answering.	Do you think Mr. Pace made a good decision?
or	This is a choice question. Choice questions are questions that offer a choice of several options as an answer. They are made up of two parts, which are connected by the conjunction “or”.	Were you working out for fun or were you into body building?

Table 1: Question dialog acts

Category	Description	Example
num	It is a what question specific to numeric quantities.	What is the age of your daughter?
hum	It is a what question specific to human beings.	What is the name of your daughter?
loc	It is a what question specific to locations.	What is the name of the city where your daughter lives?
ent	It is a what question specific to other entities.	What is the email address of your daughter?
des	It is a what question which generally ask descriptive questions.	What were you doing there at that point of time?

Table 2: wh-question dialog acts

Category	Description	Example
y	It is a category when a person answering the question means yes. The answer sentence can take various forms and the answer need not be exactly “yes”.	“yes”, “yeah”, “Of course”, “definitely it is”, “that’s right”, “I am sure”, etc.
y-d	It is a category when a person answering the binary question not only says yes but also given an explanation for this answer.	Yes. I play badminton because my doctor advised me to.
y-followup	The answer is yes, but in the answer, there is another question which pertains to the question asked.	Yes I have seen them. But what do you mean by inside the elevator?
n	It is a category when a person answering the question means no. Again, the answer need not be exactly “no”.	“No”, “I don’t think so”, “certainly not”, “I am afraid not”, etc.
n-d	It is a category when a person answering the binary question not only says no but also given an explanation for this answer.	No. I am not interested in playing Cricket because it takes a lot of time
n-followup	The answer is no, but in the answer, there is another question which pertains to the question asked.	That is not me. Do you think that is me?
sno	It is a statement which is a non-opinion. This is an informative statement made by the person answering the question.	I retired from my job in 2010.
so	It is a statement which is an opinion. It is a statement which is actually an opinion of the person answering rather than a general statement.	I believe retiring from my job was the best decision I made.
ack	It is a response which indicates acknowledgment.	“Okay”, “Um-hum”, “I see”, etc.
dno	It is a response given when the person doesn’t know, or doesn’t recall, or is unsure about the answer to the question asked.	I don’t recall what happened that day
confront	The answer contains no information. It is a confrontation by the deponent to the question asked.	So do you say that I have given you the wrong information?

Table 3: Answer dialog acts

a fully connected neural network to check for efficacy of sentence embeddings like BERT in dialog act classification. The following sections describe the different classification methods we used to classify the dialog acts.

3.2.1 Classification using CNN. Work in [22] used CNN to capture the n-gram representation of a sentence using convolution. A window size provided as a parameter was used to define the number of words to be included in the convolution filter. Figure 1 shows the convolution operation capturing a bi-gram representation. We used the architecture from the original work in [22] for learning the sentence representation using a CNN. We added a feed-forward neural network layer in front of the representation layer to finally classify the dialog act for a given sentence. Tokens from a sentence are transformed into word vectors using word2vec, and fed into the network. This is followed by the convolution and max-pooling

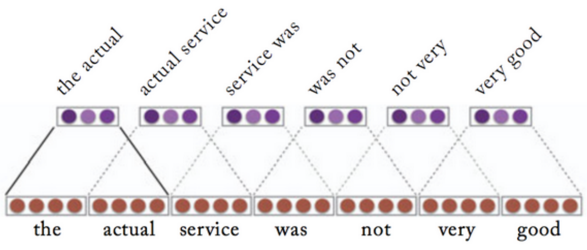


Figure 1: An n-gram convolution filter [15, 22].

operations. The final sentence has a fixed size representation irrespective of sentence length. As the system trains, the network is able to learn a sentence embedding as part of this layer. This representation is rich since it captures the semantic and syntactic relations between the words. Figure 2 shows a reference architecture of the whole CNN based approach for two classes.

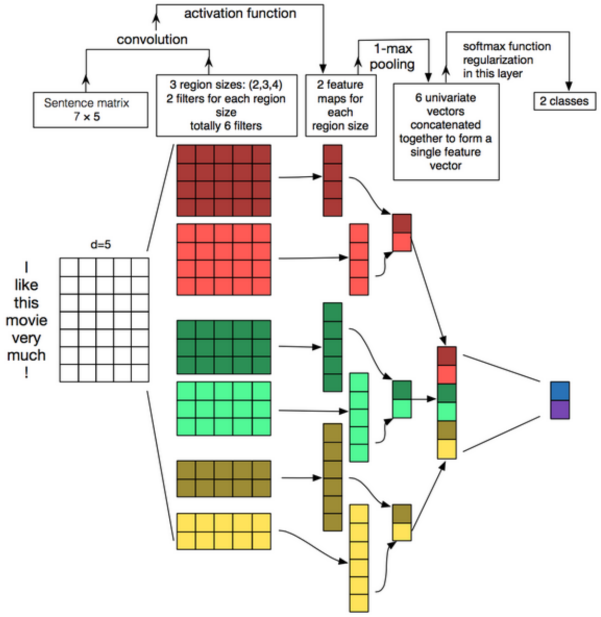

activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

Figure 2: CNN based classifier architecture [4, 22].

3.2.2 Classification using LSTM with attention. Work in [42] used a bi-directional LSTM with an attention mechanism to capture the most important information contained in a sentence. It did not use any classical NLP system based features. Even though CNN can capture some semantic and syntactic dependencies between words using a larger feature map, it struggles to capture the long term dependencies between words if the sentences are long. LSTM based network architectures are better equipped to capture these long term dependencies since they employ a recurrent model. The context of the initial words can make their way down the recurrent chain based on the activation of the initial words and their gradients, during the back propagation phase.

Figure 3 shows the network architecture of the system. The words are fed into the network using their vector representation. The network processes the words in both directions. This helps the network learn the semantic information not only from the words in the past, but also from the words in the future. The output layers of both the directional LSTMs are combined as one, using an element-wise sum. An attention layer is added to this combined output, with coefficients for each output unit. These coefficients act as the attention mechanism; attention priorities are learned by the system during the training phase. These coefficients capture the relative importance of the terms in the input sentence. The word embeddings were also learned as part of the training. Dropout [35] was applied to the embedding, LSTM, and penultimate layers. L2-norm based penalties were also applied as part of the regularization.

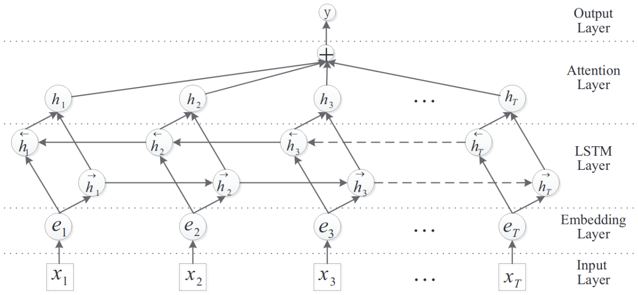

y
Output Layer
Attention Layer
LSTM Layer
Embedding Layer
Input Layer
h_1 h_2 h_3 ... h_T
$\overleftarrow{h}_1$ $\overleftarrow{h}_2$ $\overleftarrow{h}_3$ $\overleftarrow{h}_T$
$\overrightarrow{h}_1$ $\overrightarrow{h}_2$ $\overrightarrow{h}_3$ $\overrightarrow{h}_T$
e_1 e_2 e_3 ... e_T
x_1 x_2 x_3 ... x_T

Figure 3: Bi-directional LSTM with attention architecture [42].

3.2.3 Classification using BERT. In this method, we generate the sentence embeddings of the questions and answers via the BERT pre-trained model. BERT can be fine-tuned to any NLP task by adding a layer on the top of this architecture which makes it suitable for the task. Figure 4 shows the high-level architecture consisting of various components like embeddings and transformers.

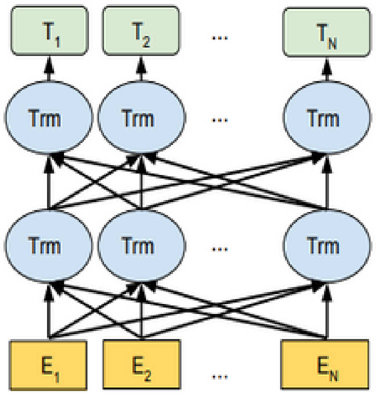


Figure 4: BERT architecture [11, 37].

In our system implementation, we used the BERT reference architecture and added a feed-forward neural network layer on top of BERT sentence embeddings. We want to classify text with length that varies from roughly a portion of one sentence to a large paragraph. Further, we are performing a single sentence classification and not a sentence pair classification, as was mentioned in the BERT paper. We use the BERT-Base, Cased pre-trained model for our classification esperiments. Figure 5 shows the architecture for our classifier.

In our experiment section, we will refer to the introduced classification methods as CNN, Bi-LSTM, and BERT, respectively.

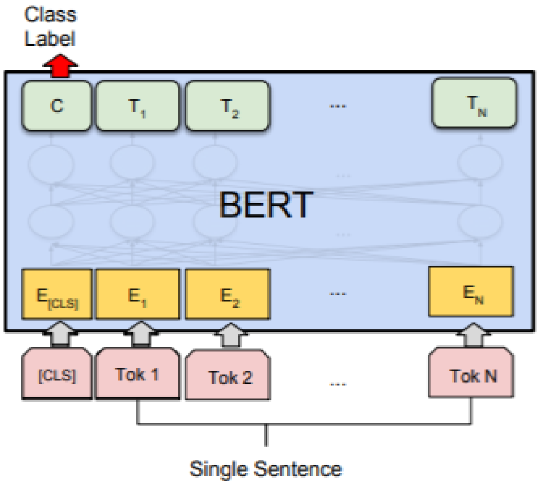


Figure 5: BERT single sentence classification architecture[11].

4 DATASET

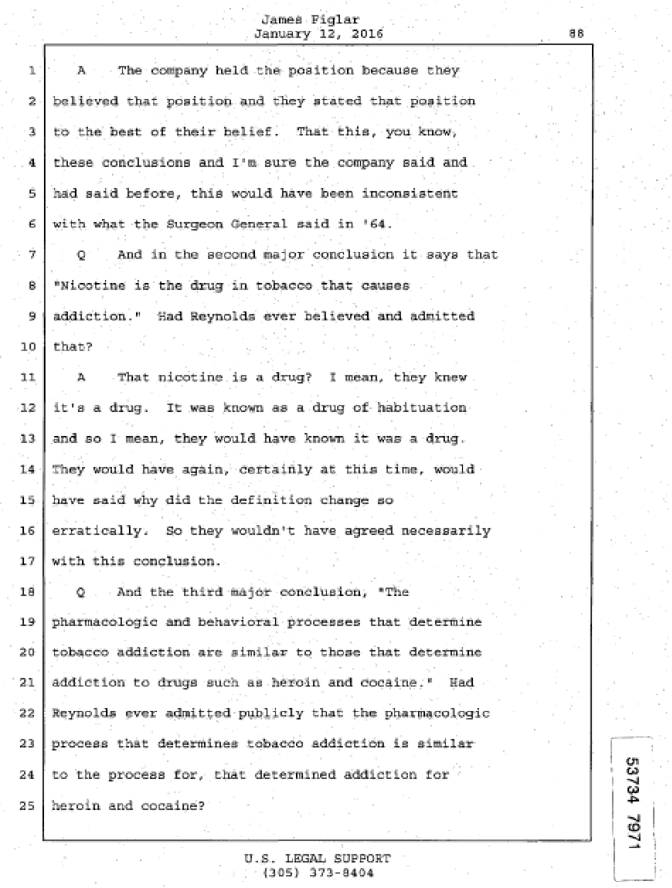
Legal depositions and trial testimonies represent a type of conversations which have a specific format, where the attorney asks questions and the deponent or witness answers those questions. Figure 6 shows an example of a page in a legal deposition. Proper parsing of legal depositions is necessary to perform analysis for downstream tasks like summarization.

4.1 Proprietary Dataset

For our dialog acts classification experiments, we performed all our work on a proprietary dataset, provided by Mayfair Group LLC. This dataset was made available to us as a courtesy by several law firms. Our classification experiments were performed on this dataset and results of this paper reflect the same. This dataset consists of around 350 depositions. The format of these documents follows conventional legal deposition standards.

4.2 Tobacco Dataset

The roughly 14 million Truth Tobacco Industry Documents constitutes a public dataset, which contains legal documents, related to the settlement of court cases between US states and the seven major tobacco industry organizations, on willful actions of tobacco companies to sell tobacco products despite their knowledge of the harmful effects. It was created in 2002 by the UCSF Library and Center for Knowledge Management to provide public access to the many legal documents related to that settlement. This dataset includes around 12,300 publicly available legal deposition documents which can be accessed from the website maintained by UCSF [24]. Our analysis and results can also be reproduced on this publicly available dataset.



James Figlar
January 12, 2016 88

A The company held the position because they believed that position and they stated that position to the best of their belief. That this, you know, these conclusions and I'm sure the company said and had said before, this would have been inconsistent with what the Surgeon General said in '64.

Q And in the second major conclusion it says that "Nicotine is the drug in tobacco that causes addiction." Had Reynolds ever believed and admitted that?

A That nicotine is a drug? I mean, they knew it's a drug. It was known as a drug of habituation and so I mean, they would have known it was a drug. They would have again, certainly at this time, would have said why did the definition change so erratically. So they wouldn't have agreed necessarily with this conclusion.

Q And the third major conclusion, "The pharmacologic and behavioral processes that determine tobacco addiction are similar to those that determine addiction to drugs such as heroin and cocaine." Had Reynolds ever admitted publicly that the pharmacologic process that determines tobacco addiction is similar to the process for, that determined addiction for heroin and cocaine?

U.S. LEGAL SUPPORT
(305) 373-8404

53734 7971

Source: https://www.industrydocuments.ucsf.edu/docs/psmw0225

Figure 6: Example page of a deposition.

Due to client privacy and confidentiality concerns, we are unable to share the proprietary dataset. The annotated tobacco dataset is available publicly[1] for the research community to use.

4.3 Data pre-processing

Legal depositions can be in a wide variety of formats like .pdf, .docx, .rtf, .txt, etc. Implementing a separate functionality for parsing different formats can be difficult and time-consuming. So, a common platform which can be used to parse deposition transcripts across all the formats in a generalized way is needed. Apache Tika [29], developed by the Apache Software Foundation, can be used to extract metadata and content from across hundreds of file types through a single interface. Apache Tika has Python support through a library called tika.

Though there is a standard format for deposition documents, different challenges were encountered while parsing the documents. Challenges faced in legal deposition document parsing include:

(1) Varying number of columns per page,
(2) Header and footer elimination, and

[1] The dataset can be downloaded from https://github.com/saurabhc123/asail_dataset

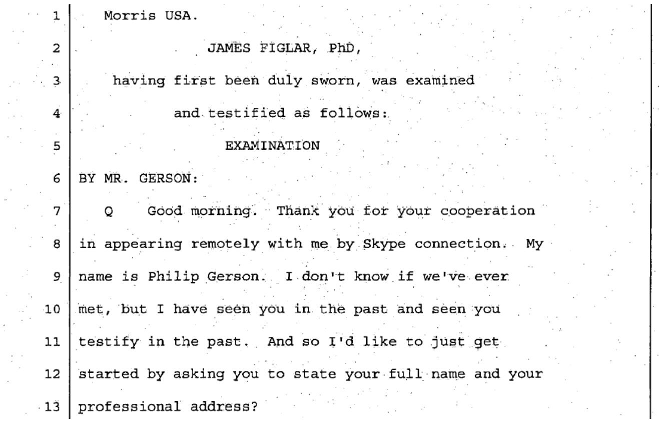
(3) Determining the starting and ending points of the actual deposition conversation within the entire document.

Generally, the PDF versions of legal depositions have multiple columns per page. Apache Tika reads multiple columns in a page separately by recognizing column separations which are encoded as extended ASCII codes. Hence, text from separate columns are parsed in the correct sequence.

Header and footer in legal depositions constitute several things like the name of the person being deposed, name of the attorney, name of the law firm, e-mail IDs, phone numbers, page numbers, etc. Figure 6 shows an example of a page in a legal deposition with header and footer. We read the content parsed by Apache Tika line by line and use regular expressions (regex) in Python to search for a pattern within each line of the text. Using regex in Python, we convert every line to a string which contains only alphabets, periods, and question marks. Then, we use a dictionary in Python to store all the patterns and the list of indices of the lines in which those pattern has appeared. Finally, we check for the patterns which satisfy the below constraints and remove those lines from the text.

(1) The number of times these patterns appear must be greater than or equal to the number of pages of the document.
(2) Those lines must not begin with the answer or question tags ('A.' and 'Q.') and must not end with a question mark.

For example, in the document which is represented by Figure 6, patterns "sourcehttpswww.industrydocuments.ucsf.edudocspsmw", "january", "jamesfiglar", "u.s.legalsupport" satisfy all of the above constraints, and hence the lines containing these patterns are removed from the entire text with the help of their indices which are stored in the dictionary.



Morris USA.

JAMES FIGLAR, PhD,

having first been duly sworn, was examined

and testified as follows:

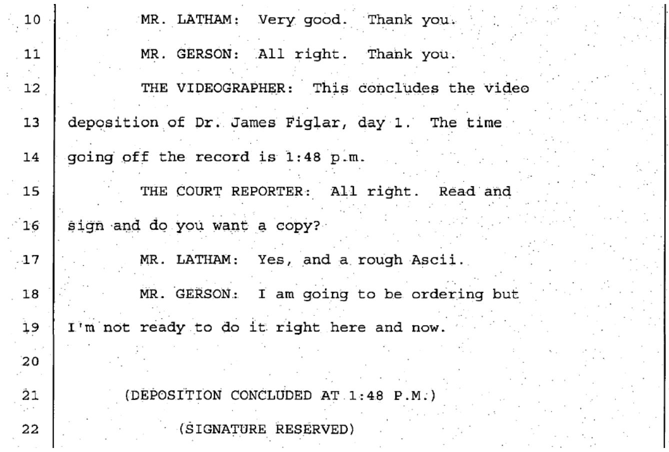
EXAMINATION

BY MR. GERSON:

Q Good morning. Thank you for your cooperation in appearing remotely with me by Skype connection. My name is Philip Gerson. I don't know if we've ever met, but I have seen you in the past and seen you testify in the past. And so I'd like to just get started by asking you to state your full name and your professional address?

Figure 7: Example of beginning of "EXAMINATION" segment.

After cleaning the text, pre-processing of data had to be done to extract the needed data in the required format. A deposition transcript can contain multiple segments within it (like "INDEX", "EXHIBITS", "APPEARANCES", "EXAMINATION", "STIPULATIONS", "CERTIFICATIONS", etc). For our work, we only needed the "EXAMINATION" segment where the actual conversation between attorney(s) and deponent takes place. Figures 7 and 8 represent the starting and ending of the "EXAMINATION" segment. We only extract the "EXAMINATION" segment based on the observed patterns that represent beginning and ending of this segment that hold across our various depositions.



MR. LATHAM: Very good. Thank you.

MR. GERSON: All right. Thank you.

THE VIDEOGRAPHER: This concludes the video deposition of Dr. James Figlar, day 1. The time going off the record is 1:48 p.m.

THE COURT REPORTER: All right. Read and sign and do you want a copy?

MR. LATHAM: Yes, and a rough Ascii.

MR. GERSON: I am going to be ordering but I'm not ready to do it right here and now.

(DEPOSITION CONCLUDED AT 1:48 P.M.)

(SIGNATURE RESERVED)

Figure 8: Example of ending of "EXAMINATION" segment.

Finally, our pre-processing methods removed the noise from the text and only extracted the conversation part of the deposition.

5 EXPERIMENTAL SETUP AND RESULTS

5.1 Experimental Setup

The overall size of the derived dataset developed from the public dataset for dialog acts classification was a total of about 2500 questions and answers. This entire dataset was manually annotated, to provide a ground truth for evaluation. The dataset then was randomly divided into train, validation, and test datasets in the ratio 70:20:10, respectively, to be studied using each of the three classifiers. Table 4 shows the distribution of the classes for the whole dataset.

Class	Counts	% of Total
ack	36	1.46
bin	437	17.67
bin-d	369	14.92
cc	0	0.00
co	4	0.16
confront	21	0.85
dno	142	5.74
n	76	3.07
n-d	74	2.99
n-followup	1	0.04
nu	29	1.17
or	18	0.73
qo	25	1.01
sno	567	22.93
so	25	1.01
wh	298	12.05
wh-d	57	2.3
y	226	9.14
y-d	66	2.67
y-followup	2	0.08
Total	2473	-

Table 4: Class distribution for the dataset

5.1.1 Environment setup. All the classification experiments were run on a Dell server running Ubuntu 16.04, with 32 GB RAM and two Tesla P40 NVIDIA GPUs.

5.1.2 CNN classifier. Parameters that were fine-tuned for the CNN with word2vec embeddings classifier are:

1. hidden layer size: This was varied from 100 to 500 in steps of 100.
2. dropout: This was varied from 0.1 to 0.5 in steps of 0.1.
3. output layer activation function: sigmoid, tanh, and relu.
4. n-gram: window size base on unigram, bi-gram, and tri-gram groupings.
5. max-sequence length: It was kept constant at 32.
6. batch-size: It was kept constant at 100.
7. number of epochs: It was varied from 10 to 50 until the validation accuracy stopped improving any further.

5.1.3 LSTM classifier. Parameters that were fine-tuned for the Bi-directional LSTM with attention classifier are:

1. hidden layer size: This was varied between the values 32, 64, 128, and 256.
2. embedding size: This was varied between the values 32, 64, 128, and 256.
3. learning rate: This was varied between the values 0.0001, 0.001, 0.01, and 0.1.
4. max-sequence length: It was kept constant at 32.
5. batch-size: It was kept constant at 100.
6. number of epochs: It was varied from 10 to 50 until the validation accuracy stopped improving any further.

5.1.4 BERT classifier. Parameters that were fine-tuned for the BERT single sentence classifier are:

1. learning rate: This was varied between the values 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, and 0.1.
2. max-sequence length: It was kept constant at 32.
3. batch-size: It was kept constant at 100.
4. number of epochs: It was varied from 10 to 50 until the validation accuracy stopped improving any further.

5.2 Results

5.2.1 System Comparisons. Table 5 lists each of the three classifiers and their corresponding best test F1 score. BERT outperformed the other methods by a significant margin and achieved an F1 score of 0.84.

Classifier	F1-score
BERT	**0.84**
CNN	0.57
LSTM	0.71

Table 5: Classifiers and their F1 scores. Best result in bold.

Tables 6, 7, and 8 gives the parameters of the CNN, LSTM and BERT classifiers, respectively, with which the best results were achieved.

Parameters	Values
hidden layer size	200
dropout	0.5
output layer activation function	sigmoid
n-gram	trigram
max-sequence length	32
batch-size	100
number of epochs	30

Table 6: Best fine tuned parameters for CNN classifier for Tobacco dataset

Parameters	Values
hidden layer size	128
embedding size	256
learning rate	0.01
max-sequence length	32
batch-size	100
number of epochs	30

Table 7: Best fine tuned parameters for LSTM classifier for Tobacco dataset

Parameters	Values
learning rate	2e-5
max-sequence length	32
batch-size	100
number of epochs	30

Table 8: Best fine tuned parameters for BERT classifier for Tobacco dataset

Figures 9, 10, and 11 represent train and test accuracy across number of epochs for the CNN, LSTM, and BERT classifiers, respectively.

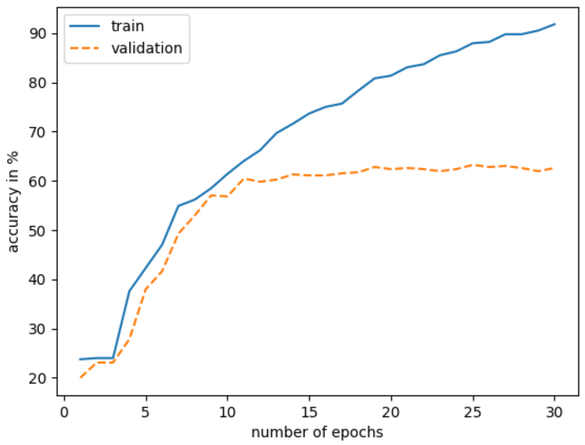


Figure 9: Train & test accuracy vs. epochs for CNN

We observe from Figures 9, 10, and 11 that after 15 epochs, the training accuracy is still increasing but the validation accuracy remains almost constant. This indicates that after 15 epochs, the

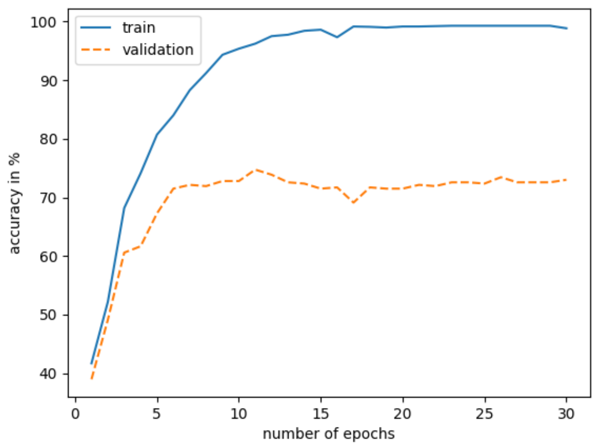


Figure 10: Train & test accuracy vs. epochs for LSTM

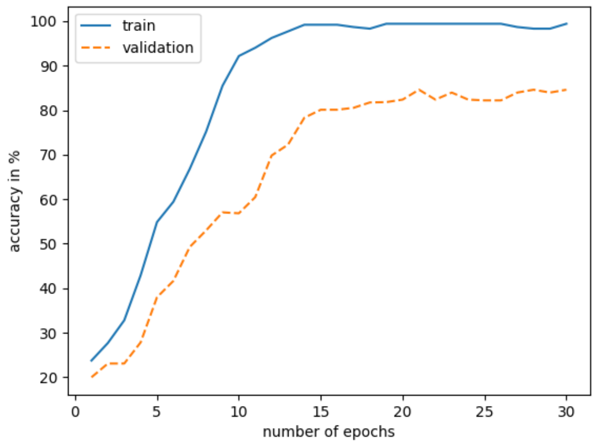


Figure 11: Train & test accuracy vs. epochs for BERT

models achieve a good fit. We also observe that the validation accuracy of BERT is highest compared to the CNN and LSTM classifiers, reaching around 83%. This is another indicator that the BERT classifier is best suited for dialog acts classification of legal depositions as compared to the CNN and LSTM classifiers.

Table 9, 10 and 11 show the precision, recall and the F1 scores for the CNN, LSTM, and BERT classifiers respectively.

5.3 Error Analysis

We chose the best performing classification results and performed a detailed error analysis on the misclassifications. Table 12 discusses the errors associated with each dialog act. We have not included the dialog acts that had fewer than 3 test samples or misclassifications.

Class	Precision	Recall	F1-score
ack	1.00	0.67	0.80
bin	0.48	0.48	0.48
bin-d	0.75	0.71	0.73
confront	0.00	0.00	0.00
dno	0.62	0.50	0.55
n	1.00	0.70	0.82
n-d	1.00	0.40	0.57
nu	0.00	0.00	0.00
or	0.00	0.00	0.00
qo	0.00	0.00	0.00
sno	0.51	0.80	0.62
so	0.00	0.00	0.00
wh	0.44	0.50	0.47
wh-d	0.00	0.00	0.00
y	0.84	0.91	0.87
y-d	1.00	0.67	0.80
avg / total	0.57	0.60	0.57

Table 9: Classification scores for CNN

Class	Precision	Recall	F1-score
ack	0.86	1.00	0.92
bin	0.79	0.73	0.76
bin-d	0.67	0.83	0.74
confront	0.50	0.50	0.50
dno	0.82	0.75	0.78
n	1.00	0.75	0.86
n-d	0.80	0.80	0.80
nu	0.00	0.00	0.00
or	0.00	0.00	0.00
qo	0.00	0.00	0.00
sno	0.61	0.74	0.67
so	0.50	0.25	0.33
wh	0.88	0.82	0.85
wh-d	0.50	0.14	0.22
y	0.70	0.89	0.78
y-d	1.00	0.55	0.71
avg / total	0.72	0.72	0.71

Table 10: Classification scores for LSTM

Class	Precision	Recall	F1-score
ack	1.00	1.00	1.00
bin	0.78	0.93	0.85
bin-d	0.74	0.74	0.74
confront	1.00	0.50	0.67
dno	0.93	1.00	0.97
n	0.89	1.00	0.94
n-d	0.86	0.75	0.80
nu	0.33	1.00	0.50
or	0.00	0.00	0.00
qo	0.00	0.00	0.00
sno	0.91	0.84	0.87
so	0.50	1.00	0.67
wh	0.92	0.85	0.88
wh-d	0.62	0.56	0.59
y	0.92	1.00	0.96
y-d	0.92	0.86	0.89
avg / total	0.83	0.84	0.84

Table 11: Classification scores for BERT

6 CONCLUSION AND FUTURE WORK

We parsed legal depositions in a wide variety of formats and extracted the necessary conversation information, also removing

Class	Analysis
bin	There were certain cases that were classified as bin-d instead of bin. There is a very subtle difference between bin and bin-d and the classifier sometimes struggles to detect this subtlety. There were a couple of instance where bin was classified as “wh”. This happened because the question had the word how or what included in it, but it was framed in such a way that the response would be a yes or no. “Is Altria Group, Inc. what is considered to be a public company?”. In this question, the classifier is not able to distinguish the exact difference and is classifying based on the observed words.
bin-d	Most of the misclassifications in this dialog act was the assignment to the “bin” category. The classifier is taking cues from the word and sometimes fails to recognize that there is some context to the question before the actual question is being asked.
confront	Most of the classifications of this kind were erroneous. This is due to the lack of training data for the classifier to effectively learn to distingish the “confront” class from the other classes. There are very few instances of this class in the depositions. Adding more specific training data for this class would help increase the classification performance.
dno	The misclassifications for the this class was due to the semantic formation of the sentence and there is very little for the classifier to distinguish from the other classes of “n-d” and “n”.
n-d	The few misclassications for this class was a result of having the word “no” appended in addition to a response of a “n-d” kind.
qo	Lack of training data and very few distinguishing words for the classifier to make an accurate judgment. More training data for this class would help increase the classification performance.
sno	The misclassifications for this class had the class assignment to “bin-d” or “wh”. On further analysis it was observed that the misclassified sentences were very long in length. Some of them ended with a form that made the classifier assign them in the “bin-d” or the “wh” classes.
so	There is very little to distinguish a “so” class from a “sno” class. Most misclassifications were of this kind. We believe they can be merged into one single category as part of our future work.
wh	The misclassifications for this class involved the assignment of the statement to the wh-d class. Looking at the statements, we can conclude that those statements could belong to the “wh-d” class. This was more of an annotation error instead of a misclassification.
y-d	In the two misclassifications, one of the statements was too long and was assigned the “so” category. For the other instance, the presence of the word “yes” in the statement made it get assigned to the “y” category, even though there was a sentence preceeding it.

Table 12: Error analysis

much of the noise, allowing for natural language processing (NLP) and deep learning techniques to be employed for further processing.

State-of-the-art summarization methods and NLP techniques are difficult to apply to question-answer pairs. Our preliminary testing with summarization methods applied to QA pairs led to poor results. Hence we desire a semantically equivalent, grammatically correct, and linguistically fluent representation to replace each QA pair. This should retain key information from the QA pair so that summaries generated from that representation do not lose any important information from the actual conversation. To achieve this, we carefully defined and developed a dialog act ontology which contains 20 dialog acts to capture the intention of the speaker behind the utterance. The quality of the set of dialog acts is also enriched based on our study of the legal deposition domain. Classification of each question and answer into these dialog acts should aid in developing specific NLP rules or techniques to convert each question-answer pair into an appropriate representation. For classification purposes, we have created our own dataset by manually annotating around 2500 questions and answers into their corresponding dialog acts. This dataset helped us in training the classifiers and also in evaluating the performance of the classifiers.

We have developed three deep learning based classification methods for dialog acts classification:

- Convolutional Neural Network (CNN) with word2vec embeddings,
- Bi-directional Long Short Term Memory (LSTM) with attention mechanism, and
- Bidirectional Encoder Representations from Transformers (BERT).

We experimented with these three classifiers and fine-tuned their various parameters. We performed training, validation, and testing with each of the three classifiers. We achieved F1 scores of 0.57 and 0.71 using the CNN and the LSTM based classifiers, respectively. The highest F1 score of 0.84 was achieved using the BERT sentence embeddings based classifier on the dialog act classification task.

We plan to extend this work in the following ways.

(1) Use context information for Dialog Acts classification such as using the dialog acts from previous utterances [3] to classify the current dialog act, to improve the classification accuracy.
(2) Develop NLP and deep learning techniques to convert a question-answer pair to a semantically equivalent representation, to which it will be easy to apply a variety of NLP tools.
(3) Use state-of-the-art deep learning based abstractive summarization methods to generate summaries from those representations.
(4) Develop explainable AI methods so it will be clear how summaries were generated.

ACKNOWLEDGMENTS

This work was made possible by the Virginia Tech’s Digital Library Research Laboratory (DLRL). We would also like to thank Ashin Marin Thomas for her help with data annotation and running the experiments. Data in the form of legal depositions was provided by Mayfair Group LLC. In accordance with Virginia Tech policies and procedures and my ethical obligation as a researcher, 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] Jeremy Ang, Yang Liu, and Elizabeth Shriberg. Automatic dialog act segmentation and classification in multiparty meetings. In *Proceedings.(ICASSP’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 1, pages I–1061. IEEE, 2005.

[2] Phil Blunsom, Edward Grefenstette, and Nal Kalchbrenner. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. ACL, 2014.

[3] Chandrakant Bothe, Cornelius Weber, Sven Magg, and Stefan Wermter. A context-based approach for dialogue act recognition using simple recurrent neural networks. In *Proceedings of the Eleventh International Conference on Language*

Resources and Evaluation (LREC-2018), 2018.

[4] Denny Britz. Understanding convolutional neural networks for NLP. *URL: http://www. wildml. com/2015/11/understanding-convolutional-neuralnetworks-for-nlp/(visited on 11/07/2015)*, 2015.

[5] Eduardo PS Castro, Saurabh Chakravarty, Eric Williamson, Denilson Alves Pereira, and Edward A Fox. Classifying short unstructured data using the Apache Spark platform. In *Proceedings of the 17th ACM/IEEE Joint Conference on Digital Libraries*, pages 129–138. IEEE Press, 2017.

[6] Lin Chen and Barbara Di Eugenio. Multimodality and dialogue act classification in the RoboHelper project. In *Proceedings of the SIGDIAL 2013 Conference*, pages 183–192, 2013.

[7] Kyunghyun Cho, Bart Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[8] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua 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*, pages 103–111, 2014.

[9] William Coster and David Kauchak. Simple English Wikipedia: a new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 665–669. Association for Computational Linguistics, 2011.

[10] Misha Denil, Alban Demiraj, Nal Kalchbrenner, Phil Blunsom, and Nando de Freitas. Modelling, visualising and summarising documents with a single convolutional neural network. *arXiv preprint arXiv:1406.3830*, 2014.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[12] Alfred Dielmann and Steve Renals. Recognition of dialogue acts in multiparty meetings using a switching DBN. *IEEE transactions on audio, speech, and language processing*, 16(7):1303–1314, 2008.

[13] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[14] Raul Fernandez and Rosalind W Picard. Dialog act classification from prosodic features using support vector machines. In *Speech Prosody 2002, International Conference*, 2002.

[15] Yoav Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309, 2017.

[16] Simon Haykin. *Neural networks*, volume 2. Prentice Hall New York, 1994.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] Gang Ji and Jeff Bilmes. Dialog act tagging using graphical models. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 1, pages I–33. IEEE, 2005.

[19] Daniel Jurafsky, Elizabeth Shriberg, Barbara Fox, and Traci Curl. Lexical, prosodic, and syntactic cues for dialog acts. *Journal on Discourse Relations and Discourse Markers*, 1998.

[20] Nal Kalchbrenner and Phil Blunsom. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126, 2013.

[21] Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. Classifying dialogue acts in one-on-one live chats. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 862–871. Association for Computational Linguistics, 2010.

[22] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746âĂŞ–1751. ACL, 2014.

[23] Pavel Král and Christophe Cerisara. Automatic dialogue act recognition with syntactic features. *Language resources and evaluation*, 48(3):419–441, 2014.

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

[25] Yang Liu. Using SVM and error-correcting codes for multiclass dialog act classification in meeting corpus. In *Ninth International Conference on Spoken Language Processing*, 2006.

[26] Yang Liu, Kun Han, Zhao Tan, and Yun Lei. Using context information for dialog act classification in DNN framework. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2170–2178, 2017.

[27] Mark Davies. Google books corpora, 2011. [Online; accessed 28-April-2019].

[28] Marion Mast, Ralf Kompe, Stefan Harbeck, Andreas Kießling, Heinrich Niemann, Elmar Noth, Ernst Günter Schukat-Talamazzini, and Volker Warnke. Dialog act classification with the help of prosody. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, volume 3, pages 1732–1735. IEEE, 1996.

[29] Chris Mattmann and Jukka Zitting. *Tika in Action*. Manning Publications Co., Greenwich, CT, USA, 2011.

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[31] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[32] Silvia Quarteroni, Alexei V Ivanov, and Giuseppe Riccardi. Simultaneous dialog act segmentation and classification from human-human spoken conversations. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5596–5599. IEEE, 2011.

[33] LM Rojas-Barahona, M Gašić, N Mrkšić, PH Su, S Ultes, TH Wen, and S Young. Exploiting sentence and context representations in deep neural models for spoken language understanding. In *COLING 2016-26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers*, pages 258–267, 2016.

[34] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 101–110. ACM, 2014.

[35] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[36] Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373, 2000.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[38] Anand Venkataraman, Luciana Ferrer, Andreas Stolcke, and Elizabeth Shriberg. Training a prosody-based dialog act tagger from unlabeled data. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE, 2003.

[39] Xin Wang, Yuanchao Liu, SUN Chengjie, Baoxun Wang, and Xiaolong Wang. Predicting polarities of tweets by composing word embeddings with long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1343–1353, 2015.

[40] N Webb, M Hepple, and Y Wilks. Dialog act classification based on intra-utterance features. cs-05-01. *Dept. of Computer Science, University of Sheffield, UK*, 2005.

[41] Jason 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)*, pages 23–24, 2012.

[42] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212, 2016.