Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 590–599
Marseille, 11–16 May 2020

Augmenting Small Data to Classify Contextualized Dialogue Acts for Exploratory Visualization

Abhinav Kumar, Barbara Di Eugenio, Jillian Aurisano, Andrew Johnson
University of Illinois at Chicago
Chicago, IL USA
{akumar34, bdieugen, jauris2, ajohnson}@uic.edu

Abstract

Our goal is to develop an intelligent assistant to support users explore data via visualizations. We have collected a new corpus of conversations, *C*HICAGO-CRIME-VIS, geared towards supporting data visualization exploration, and we have annotated it for a variety of features, including contextualized dialogue acts. In this paper, we describe our strategies and their evaluation for dialogue act classification. We highlight how thinking aloud affects interpretation of dialogue acts in our setting and how to best capture that information. A key component of our strategy is data augmentation as applied to the training data, since our corpus is inherently small. We ran experiments with the Balanced Bagging Classifier (BAGC), Conditional Random Field (CRF), and several Long Short Term Memory (LSTM) networks, and found that all of them improved compared to the baseline (e.g., without the data augmentation pipeline). CRF outperformed the other classification algorithms, with the LSTM networks showing modest improvement, even after obtaining a performance boost from domain-trained word embeddings. This result is of note because training a CRF is far less resource-intensive than training deep learning models, hence given a similar if not better performance, traditional methods may still be preferable in order to lower resource consumption.

Keywords: Dialogue, Corpus, Statistical and Machine Learning Methods

1. Introduction

The role of context on interpretation has been recognized since the very beginning of NLP. This is even more important in dialogue processing, where determining the intent behind the interlocutor's utterance is paramount to appropriately acting on that intent.

The underlying intent is often captured via the notion of a speech act (Searle, 1975), commonly operationalized as a dialogue act (DA) in NLP. Most methods for DA classification, whether earlier sequential models (Stolcke et al., 2000; Tavafi et al., 2013) or more recent deep learning models (Kumar et al., 2018; Ahmadvand et al., 2019) operate on conversations consisting of adjacency pairs (Schegloff and Sacks, 1973), where the DA by speaker 1 strongly conditions the possible responses by speaker 2.

However, such dependencies are not necessarily preserved in other kinds of conversation, for example asynchronous conversations such as email or online forums (Tavafi et al., 2013); or conversations where speakers solve a complex problem, and where multi-utterance turns may include, within the turn itself, a richer context for the intent that is communicated.

Our ultimate goal is to develop an intelligent assistant that supports exploration of datasets via visualizations on large displays through a natural interaction with the user. Making effective use of visualizations to make sense of complex data, is cognitively very demanding, especially for novices (Grammel et al., 2010; Brehmer and Munzner, 2013). To enable such a goal, as described in our previous work (Aurisano et al., 2015; Aurisano et al., 2016; Kumar et al., 2016; Kumar et al., 2017), we collected interactions between 16 subjects and a Visualization Expert (VE) where the subject explores Chicago crime data in order to better deploy police officers. We invited subjects to *think aloud*, to gain insight into their data exploration strategies (Van Someren et al., 1994; Popov et al., 2011; Reda et al., 2014) (our think alouds are not meta-cognitive reflections on their thought processes, but an explicit trail of their reasoning on the data). Hence, many turns by the users begin with any number of utterances pertaining to the current state of the exploration. Only then, does the user convey a new *actionable request* (AR) to the VE, who responds non-verbally, by manipulating an existing visualization or creating a new one (please see example in Figure 1, to be discussed shortly).

Our contributions are as follows. (1) We believe we are among the first to provide a model that takes into account think aloud as a local context for the interpretation of dialogue acts. We describe how we modeled DAs within specially coded constructs in our corpus, referred to as *contextualized actionable requests* (CARs). (2) Contrary to much current work on dialogue corpora, we have to face the issue of small size. This is an inherent challenge of datasets collected within contexts in which users interact with an external physical or software environment to accomplish complex goals; these datasets are often extremely time- and resource-intensive to collect. Small size of corpora is of particular concern today, since neural network models have achieved state-of-the-art performance for a variety of NLP tasks, but require vast datasets to be properly trained. Hence, we propose data augmentation, that although not new per se, has not been applied to dialogue act classification as far as we know. We show that data augmentation is an effective technique to improve DA classification on our dataset. (3) As far as evaluation is concerned, we show that a layered approach to DA classification for our domain is effective. Of the sequential classifiers, across all settings, CRF outperformed all others including the LSTM networks, despite the LSTM models making modest improvements when supplied with domain-trained word em-

beddings. While this may not be surprising since a neural model has many more parameters to train than a CRF, it shows that for smaller datasets not only are more traditional models like CRFs still competitive, but perhaps more importantly, are much faster to train and less energy-hungry (Strubell et al., 2019).

In the following, after discussing related work, we will present our corpus, our data augmentation pipeline, and experimental set-up. We will then discuss in detail the results of our experiments.

2. Related Work

Dialogue Act Classification. Numerous methods have been studied for DA classification; here we focus specifically on structured prediction such as MaxEnt (Ang et al.,) and CRF (Kim et al., 2010).

Recently, deep neural network models have achieved state-of-the-art results. Representative samples include (Khanpour et al., 2016; Kumar et al., 2018; Ahmadvand et al., 2019). Closest to our work, (Manuvinakurike et al., 2018) evaluated various LSTM and CNN networks in an image-editing domain. They modeled spoken conversations incrementally (word-by-word), to efficiently process the multiple fine-grained utterances by the user.

Inspired by hierarchical classification applied to questions (Li and Roth, 2002), we approach DA classification as two layers, in which the bottom layer is dependent on the classification outcome of the top layer. As far as we know, think aloud components of conversations have been occasionally studied, but have hardly been computationally modeled (Benotti, 2009); in some cases, they have been excluded from analysis, as being *self-addressed speech* that does not contribute to the conversation (Jovanovic et al., 2006).

Data Augmentation. Available public datasets on dialogue are limited to a few domains, mostly chatbots or information search. Often, the only choice is to manually build a new corpus, whose size is limited by the time and effort necessary to collect and annotate the data. Data augmentation, which applies class-preserving transformations, has been effective to enlarge datasets. Paraphrasing has been popular for data augmentation for various NLP tasks. Representative examples include semantic parsing (Campagna et al., 2019; Berant and Liang, 2014; Jia and Liang, 2016), question answering (Fader et al., 2013; Dong et al., 2017), and semantic slot filling (Yoo et al., 2018; Hou et al., 2018), but not for DA classification as far as we know.

Semantic Slot Filling. In spoken dialogue systems, semantic slot filling is tasked with identifying terms belonging to fixed slots and passing them as parameters to down-stream processing. It is common practice to treat semantic slot filling as a sequence labeling problem and apply a supervised learning method that trains on sequences (Mesnil et al., 2015; Hakkani-Tür et al., 2016; Wang et al., 2011; Vu, 2016).

We require semantic slot filling as a sub-task of our data augmentation pipeline. However, supporting supervised learning would require that our corpus be coded for semantic slots. Rather than investing significant effort to label the corpus and the large number of paraphrases in the data augmentation pipeline, we applied heuristic rules based on information captured at the word and phrase levels to automatically assign the semantic slots.

Interactive Systems for Visualization Exploration. Several recent systems facilitate exploration of data visualizations via interaction (Cox et al., 2001; Reithinger et al., 2005; Sun et al., 2010; Gao et al., 2015; Setlur et al., 2016; Hoque et al., 2017; Dhamdhere et al., 2017). However, these systems either do not support two-way communication, limit how the queries can be expressed, use less sophisticated language processing methods, do not support gesture interaction, or limit all operations to the same visualization on the screen. These are limitations we are currently working to overcome. So far, our prototype system (Kumar et al., 2016; Kumar et al., 2017) is capable of processing multimodal input (speech and pointing gestures to existing visualizations) as well as operate on a large screen display, producing new visualizations on the fly and letting the user manage multiple visualizations on the screen.

3. The CHICAGO-CRIME-VIS Corpus[1]

We built a multimodal corpus (Aurisano et al., 2015; Kumar et al., 2016; Kumar et al., 2017) via a study with 16 subjects. Each subject interacted (using speech and gesture) with a Visualization Expert (VE) to explore data visualizations on a large screen, related to crime data for Chicago (subjects could not see the VE but knew s/he was a human, not a system). The subjects were instructed to explore Chicago crime data in order to determine when and where to deploy police officers. The corpus contains 3,179 transcribed utterances covering 1,879 word types and 38,105 word tokens.

Figure 1 shows a two-turn user interaction with the system. The visualization *Vis 1* is currently on the screen when the user says utterances U_1 and U_2 followed by the AR U_3. The VE then responds by generating visualization *Vis 2* (not part of the CAR; Vis 2 appears alongside Vis 1). At this point, the user says utterances U_4 and U_5. For clarity, we use an example that includes bar charts; other frequently used visualizations in our corpus are line graphs and heat maps.

A subset of 449 utterances were coded for 8 types of DA's that we call *actionable requests* (ARs), since the VE can directly act on them (see Table 1). Intercoder agreement was $\kappa = 0.74$ (based on three coders annotating the same 4 subjects). We also annotated for referring expressions and gestures but we do not discuss them in this paper.

We have already deployed a complete pipeline that takes a sequence of individual spoken AR's and transforms each into a visualization, with good success (as confirmed by pilot user studies). In this paper, we discuss the next step, concerning the contextualized interpretation of an AR.

3.1. Contextualized Actionable Requests

We observed that often the speaker began by *thinking aloud* about the previous visualization (i.e., *conclusion* of the previous AR); then transitioned to *thinking aloud* about setting up the parameters for a new AR (i.e., *setup*); and fi-

[1] We intend to make the transcribed corpus, and the augmented data, publicly available in the future. In the meantime, it can be shared upon request.

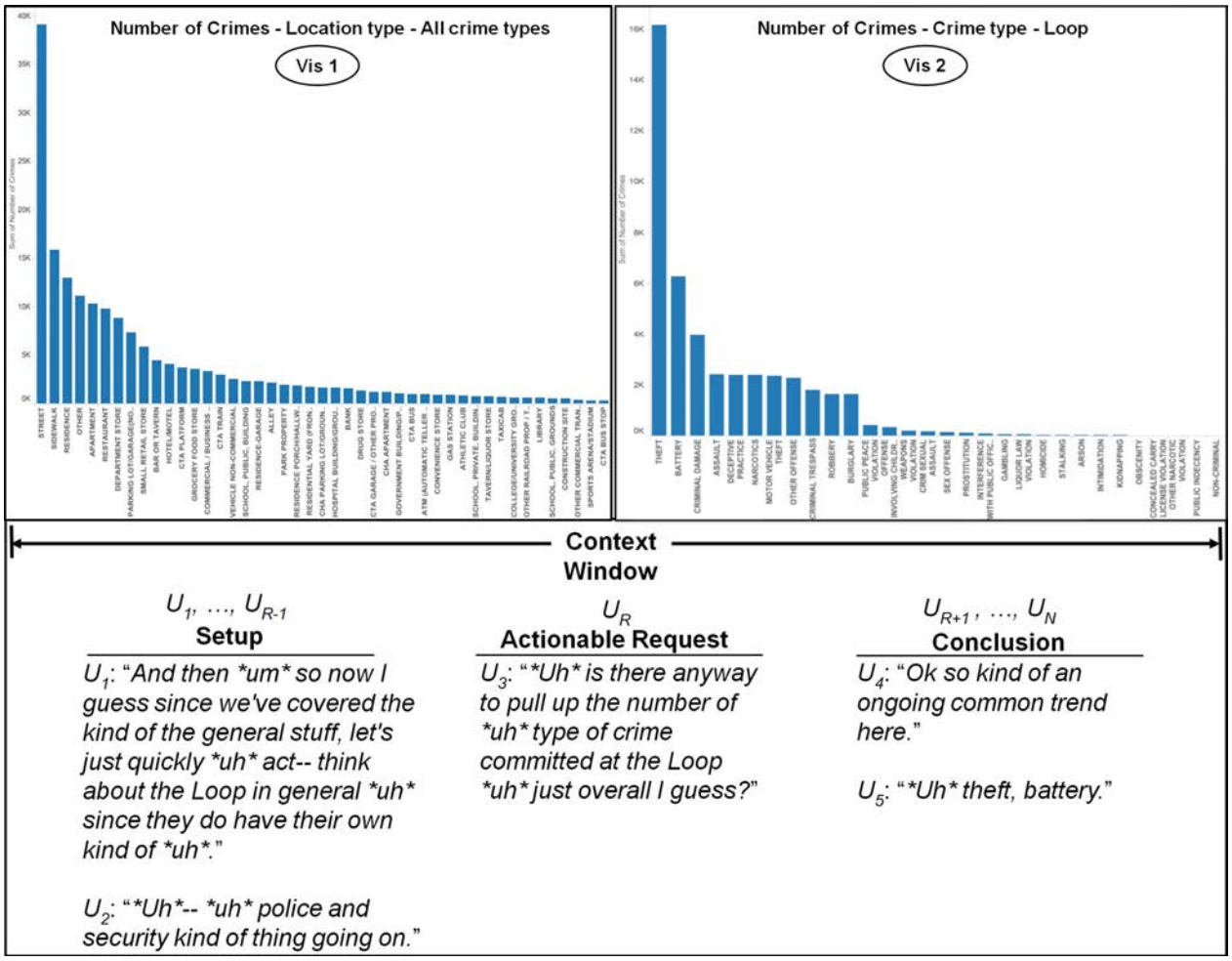

Figure 1: *Contextualized actionable requests* segment relevant user utterances in each turn of the conversation. In this sample, visualization *Vis 1* is currently on the screen when the user says utterances U_1, U_2, and U_3, the last one being an actionable request (with assigned dialogue act label *MODIFY-VIS*). The VE then responds by generating visualization *Vis 2* (not part of the CAR; Vis 2 appears alongside Vis 1). At this point, the user says utterance U_4 and U_5.

nally conveyed the AR to the VE. The VE then acted (non-verbally) to satisfy the AR by either creating a new visualization or manipulating existing ones on the screen.

We segmented the dialogues into structured constructs called CARs, each encompassing the AR, the *setup* that builds up to the AR, as well as the *conclusion* that comes after the VE responded to that AR. The CAR window size varies, with the first utterance in *setup* and final utterance in *conclusion* mentioning a data attribute while utterances in between are absent of a data attribute mention (*setup* and *conclusion* include about 1.8 and 2 utterances respectively on average). Figure 1 describes a representative example of a CAR. The corpus was coded for 449 total CARs, one for each AR, with total coverage of 1,545 out of 3,179 utterances. However, since there are only 7 *APPEARANCE* and 2 *HIGH-LEVEL-QUERY* actionable requests (see Table 1), in the following we ignore the corresponding 9 CARs; this results in a total of 440 utterances labeled as *ARs* and 1,096 labeled as *think alouds*, in the associated CARs – namely, 1,536 utterances out of 3,179. The remaining approximately 50% of the utterances are not annotated (they may cover some additional *think aloud* but not linked to any AR) and are left for future analysis. Additionally, the annotation distinguishes between CREATE-VISUALIZATION and MODIFY-VISUALIZATION by noting that in the former, a data attribute is mentioned for the first time, hence these two labels are merged in the following.

4. Approach

The conversations we collected in the *CHICAGO-CRIME-VIS* corpus differ from most dialogue corpora both because the VE's turn consists of the visualization(s) she produces (the VE often returned more than one visualization); and because of the *think aloud* component we discussed earlier. The CAR construct formalizes the fact that the user may speak any number of *think aloud* surrounding the AR proper (either as *set-up* or as *conclusion*). As such, the goal of our DA model is to tag each utterance in a given CAR with one of 6 labels: *think aloud* or one of the 5 AR types in Table 1 (merging *CREATE/MODIFY-VISUALIZATION*, and excluding *APPEARANCE* and *HIGH-LEVEL-QUERY*). Once an AR has been recognized, the surrounding *think alouds* can be directly classified as *setup* or *conclusion* according to their position wrt the AR in question. The approach we propose presupposes CARs have been segmented, and operates on

Dialogue Act	Freq	Example
CREATE-VISUALIZATION	198	*"so *uh*, can I see the visualization for crime in the Chicago neighborhoods?"*
CLARIFICATION	82	*"Ok, so, what do you mean by non-criminal?"*
WINDOW-MANAGEMENT	57	*"If you want you can close these graphs as I won't be needing it anymore"*
FACT-BASED	43	*"So, what kind of crime is what kin– what kind of crime is maximum?"*
PREFERENCE-BASED	33	*"*Uh* okay I somehow feel the 06-2 and 07-2 they are they're much clearer *uh* to visualize and understand rather than 06-1 and 07-1."*
MODIFY-VISUALIZATION	27	*"*Uh* do– can you show only the bar graph or do you have some any other way of visualizing the same?"*
APPEARANCE	7	*"*Um*, interesting, can I see labels on the data please?"*
HIGH-LEVEL-QUERY	2	*"So, according to you, which areas do I deploy the officers?"*

Table 1: *CHICAGO-CRIME-VIS* coding: 8 dialogue acts (Kumar et al., 2020 forthcoming).

transcribed data; we will come back to these points in the conclusions.

4.1. Data Augmentation Pipeline

Our corpus, comprised of 3,179 utterances, is comparable in size to other multimodal dialogues collected in a situated setting (i.e., our subjects interacted physically with a large screen display during conversation). For example, ELDERLY-AT-HOME (Chen et al., 2015) comprises 4.8K utterances, capturing dialogue interactions relating to a helper assisting an elderly person in performing daily living activities. Another example is (Katsakioris et al., 2019), which comprises of approximately 2.9K utterances pertaining to collaborative planning dialogues for autonomous underwater vehicles. The small size of such kinds of corpora could limit the ability of machine learning models to be trained effectively.

In contrast, many modern approaches to DA classification train on much larger datasets, and are either multimodal but not situated (e.g., the Augmented Multi-party Interaction (AMI) meeting corpus (Popescu-Belis and Estrella, 2007) contains transcription of over 171 meetings covering a duration of 100 total hours, but the subjects don't interact with / manipulate their physical environment) or are primarily unimodal (e.g., the Meeting Recorder Dialogue Act Corpus (MRDA) (Shriberg et al., 2004) and Switchboard (Jurafsky et al., 1997) contain 78k and 193k utterances respectively). We addressed data insufficiency in our domain by applying a data augmentation pipeline that uses paraphrasing.

(a) **Paraphrasing.** The pipeline starts by generating 20 raw paraphrases using a domain independent, pretrained model (Wieting et al., 2017). This model uses machine translation to obtain paraphrases and then trains on them using an LSTM network to learn sentence embeddings. In a small number of cases, our pipeline removes paraphrases which contain different punctuation but share the same words.

(b) **Semantic Slot Filling.** In spoken dialogue systems, semantic slot filling is tasked with identifying terms belonging to fixed slots and passing them as parameters to down-stream processing. Consider the example *"So can we also get a breakdown of the type of crimes for 10 AM?"* A better visualization can be realized for this user request by taking into consideration that *"types of crimes"* is referring to *"crime"* and *"10 AM"* is an instance of *"time"* (e.g., by not recognizing the specified time *"10 AM"* may result in a visualization with crime data across the entire 24 hours).

As we noted earlier, it would be impractical to annotate for semantic slots in order to automatically learn models, given the large increase in the number of utterances after including the paraphrases from the previous step. Instead we developed a two-phase algorithm that first suggests semantic slot tags using word-level information followed by confirmation, either by accepting as is or through adjustment of those suggestions, based on phrase-level analysis.

In this work, we use a Knowledge Ontology (KO), in the form of named entity slots and their possible values. We derived the KO from various sources, including the Chicago Data Portal[2], Encyclopedia of Chicago[3], and we manually added visualization relevant terms. This resulted in 11 semantic slots in the KO, e.g. *"CRIME"*, *"LOCATION"*, *"NEIGHBORHOOD"*, *"TIME"*, *"MONTH"*, *"VISUALIZATION"*. Finally Babelnet[4] and Wordnet[5] synsets were leveraged to increase the KO vocabulary, for a total of 1,637 terms.

- **Word-level suggestions.**
  An utterance u is first tokenized on words using NLTK [6]. Subsequently, some of the tokens of u are merged to form longer tokens, through three kinds of matches, including hyphen matches (e.g., treat *"crime-type"* as one term instead of two individual ones), regular expression matches on time-based terms (e.g. merge to make time interval as a single term, such as *"6 PM to 12*

[2]data.cityofchicago.org
[3]encyclopedia.chicagohistory.org
[4]https://babelnet.org/
[5]https://wordnet.princeton.edu/
[6]https://www.nltk.org/

AM"), and knowledge ontology (*KO*) matches (e.g., neighborhood *"River" "North"* handled as the single term *"River North"*). The final step of the algorithm is to assign a semantic slot for those tokens that matched against the KO.

- **Phrase-level confirmations.** The input is each (u, s), representing utterance *u* and suggested semantic slots s that was the output of the previous phase. The phrase confirmation processing applies heuristic rules to (u, s) based on the dependency relations between the words of certain phrases, as produced by the Spacy[7] dependency tree parser. Briefly, the heuristic rules examine handling of suggested semantic slots when dealing with compound nouns; adjective phrases associated with suggested semantic slots for terms modified by certain kinds of adjective modifiers; and appropriately resolving suggested semantic slots in prepositional phrases that contain more than one temporal noun complement. Finally, the confirmed semantic slots (without fillers) are included in the delexicalized form. This step produces up to 20 delexicalized forms, since paraphrases differing only in punctuation were removed in a previous step.

(c) **Delexicalization.** Our goal is to derive delexicalized forms of utterances where the semantic slot has been inferred: for example, *"And could you divide the chart based on criminal damage and then deceptive practices?"* would result in *"And could you divide the* [*VISUALIZATION*] *based on* [*CRIME*] *and then* [*CRIME*]." For us, this is trivial since in the output of the semantic slot filling step, we are already aware which semantic slot corresponds to each token in the utterance.

(d) **Surface Realization.** A surface realizer takes an utterance in delexicalized form, and replaces a semantic slot with a possible value, to realize a new utterance. One possible realization for the earlier delexicalized form could be *"And could you divide the plot based on theft and then battery."*

We use the corresponding possible values in the KO for each semantic slot under consideration, generating all combinations of their possible values. We only instantiate the first 3 semantic slots because the number of combinations can grow very large, for example to 15K if a delexicalized form contains the semantic slots *"CRIME"*, *"LOCATION"*, and *"VISUALIZATION"* with 63, 42, and 6 possible slot values respectively. As a final step, we randomly select three utterances from the list of combinations.

(e) **Synonym Substitution.** We also investigated the effectiveness of increasing the vocabulary size, by way of synonym substitution, which applied to the example from the previous step, would produce *"And could you split the plot found on theft and then battery."* For each utterance produced up to this point, we randomly select up to 3 eligible terms. Eligibility is based on 2 conditions: (1) the terms must not be tagged with semantic slots, and (2) the terms can only be nouns, non-auxiliary verbs, adjectives, and adverbs. We substitute each eligible term with a randomly selected Wordnet lemma associated with the synset for that term.

[7] spacy.io

5. Experimental Setup

We ran 5 different classifiers for each of our experiments. BAGC (Balance Bagging Classifier) was chosen to contrast with structured classification which takes advantage of the sequence, and because BAGC is set up to handle imbalanced datasets such as ours.

CRF was applied because it is a popular choice for DA classification (Kim et al., 2010; Tavafi et al., 2013). We included neural network models because, given their sensitivity to data insufficiency, they can better test the limits of our data augmentation pipeline. We selected architectures with LSTM at the core, since they are popular for sequential modeling, and specifically implemented LDNN (LSTM Deep Neural Network), BLDNN (Bidirectional LSTM Deep Neural Network), and CLDNN (Convolutional LSTM Deen Neural Network). LDNN is the simplest architecture. We included BLDNN and CLDNN because they have been effective respectively, to learn context from sequences in both directions, and for classifying text (CLDNNs have a convolutional layer in their architectures)

We used the Keras[8] library to implement the LSTM networks, the Sklearn-CRF Suite package[9] for CRF, and the Unbalanced-learn package[10] for BAGC. All of our results are compared to the baseline (e.g., when the data augmentation pipeline is disconnected from the system). All performance results are presented as weighted F1 scores, calculated using 5-cross validation for which we partitioned on the 16 subjects rather than on the utterances themselves (to preserve the entire conversational sequence); the augmented data was not included in the test fold.

5.1. Features

We include text features commonly used for DA classification, such as unigrams and bigrams, and POS tags, both coarse (e.g., PROPN, VERB, ADP) and fine (e.g., NNP, NN, VBZ, VBG). Further, we investigated sequential dependencies between words by including the links identified through dependency tree parsing (syn-dep); we also combined promising individual features, and hence added coarse-POS and fine-POS concatenated with unigrams (coarse-POS-unigrams and fine-POS-unigrams), and syn-dep concatenated with unigrams (syn-dep-unigrams). We also added numerical features to our analysis, as follows. Potentially, *preference-based* actionable requests have higher positive or negative sentiment so we added sentiment polarity score (using the NLTK-Sentiment[11] package) to help distinguish them from other kinds of actionable

[8] https://keras.io/

[9] https://sklearn-crfsuite.readthedocs.io/en/latest/

[10] https://imbalanced-learn.readthedocs.io/en/stable/

[11] http://www.nltk.org/howto/sentiment.html

requests. *Wh-word* counts are included because their distribution across AR types is not uniform, with e.g. *window-management* and *preference-based* hardly containing any. For *creating* or *modifying visualization* requests, we observed that users tend to use non-modal verbs more often, and hence added the corresponding counts. Finally, we added average word embedding due to its general success in various domains. We used manual experimentation to settle on an effective subset of these features for BAGC and CRF. All the LSTM networks follow the same training parameters. We used the first 40 tokens as maximum sequence length; we used the categorical cross-entropy loss function and used the ADAM optimizer (Kingma and Ba, 2014). We trained on batch size of 300 and let the networks train for 200 epochs (see below for further details).

(a) **Balanced Bagging Classifier (BAGC).** It couples bagging with a form of under-sampling that resamples all the classes except the minority class during training. This helps address class imbalance in our training data. We did not use any numerical features as in preliminary experiments we found they were not helpful. We performed all evaluations using unigrams, bigrams, coarse-POS, and fine-POS as features.

(b) **CRF.** We chose text features including unigrams, coarse-POS, fine-POS, fine-POS-unigrams; and numerical features, including average of word embeddings and semantic slot counts. We also included some context via the same features for the utterance immediately preceding and immediately following the current utterance.

(c) **LSTM Classifiers.** We manually tuned their hyperparameters. The first and final layer in all their architectures consists of a 100-dimensional embedding layer and a fully-connected layer with softmax activation respectively. The layers in between vary. We stacked 2 LSTMs (which has the advantage of learning longer dependencies over a single LSTM layer) of 100 hidden units each for LDNN and CLDNN. The BLDNN architecture has a bidirectional LSTM. The CLDNN has a 1D convolution layer with 512 filters and kernel size of 5. We also apply a dropout with *keep_prob=0.5* for the BLDNN and CLDNN classifiers.

Domain-targeted word embeddings. We used the continuous bag-of-words model (Mikolov et al., 2013) implemented in Gensim[12] to train 100-dimensional word embeddings for use by the LSTM networks (and as a feature for CRF). The embeddings were trained on various online data sources. Our web-crawling implementation extracted 8.9M articles, including: all articles found on *Chicago Sun Times*[13] related to Chicago neighborhood activities; Chicago history based articles from the *Encyclopedia of Chicago*[14]; 5 accumulated years of Chicago crime news articles from *CWB Chicago*[15]. An additional 1.2G archive of Chicago crime news articles were downloaded from the *Chicago Justice Project*[16]. Finally, we appended 3G of articles obtained on Wikipedia, by indexing it using the Woosh package[17] and then extracting articles using the terms in the *KO* as the search queries, in a breadth-first fashion, constrained to a depth of 2 hyperlinks.

Augmentation Multiplier. The *AUGX* parameter specifies the multiplicative factor increase of the training corpus size using the data augmentation pipeline (e.g., the 16 conversations comprising the *CHICAGO-CRIME-VIS* corpus would be effectively increased to 160 conversations when $AUGX = 10$). For our experiments we studied the effects of varying *AUGX*, from the baseline training size (e.g., $AUGX = 0$) comprising 1,238 utterances (note that 298 of the 1,536 utterances were removed because they contained less than four terms) and 17,816 tokens, to the maximum for $AUGX = 15$, comprising 18,570 utterances and 226,750 tokens (other values we experimented with are $AUGX = 3, 5, 10$). Larger *AUGX* values of 20, 30, and 100 resulted in degredation in performance and hence we only present results up to $AUGX = 15$.

6. Results

Data augmentation improves classification. Table 2 summarizes results for each classifier when varying *AUGX*, as well as including or excluding data augmentation pipeline components, comprised of paraphrasing (*P*); semantic slot filling, delexicalization, and surface realization (*F*); and synonym substitution (*S*). First of all, Table 2 shows that data augmentation improves performance: each classifier other than BAGC improves with respect to not using data augmentation, for each value of $AUGX > 0$, for all three data augmentation settings. That data augmentation is better than no data augmentation is statistically significant according to the sign test ($p < 0.001$ for each of P/P+F/P+F+S)[18].

CRF consistently performed the best for every combination of setting (P/P+F/P+F+S) and value of $AUGX$, including $AUGX = 0$ (please see the CRF rows in Table 2). We cannot conclude that pairwise differences between CRF and any other classifier's performance are statistically significant. However, CRF's general superior performance is statistically significant according to the sign test: for each setting P/P+F/P+F+S, CRF performs better ($p < 0.0001$).[19]

As expected, BAGC performed the worst, likely because the classifier is not set up to capture sequential dependencies among the words and utterances. The LSTM networks

[12]https://radimrehurek.com/gensim/

[13]https://chicago.suntimes.com/section/the-grid/

[14]http://www.encyclopedia.chicagohistory.org/

[15]http://www.cwbchicago.com/

[16]https://chicagojustice.org/

[17]https://whoosh.readthedocs.io/en/latest/

[18]The sign test assesses whether the number M of successes out of N trials is due to chance or not. In our case, each run of a classifier with $AUGX > 0$ is a trial; it is considered as a success if performance is higher than with $AUGX = 0$. In Table 2, each pipeline setting P/P+F/P+F+S, results in 20 trials (number of classifiers times number of $AUGX > 0$ values); the number of successes ranges from a minimum of 17 (P+F+S, $p = 0.0013$) to a maximum of 19 (P+F, $p < 0.0001$).

[19]In this case, a trial is a classifier different from CRF, and a success is whether the CRF performance for that setting and $AUGX$ value, is higher.

Method	AUGX = 0	3	5	10	15
	P				
BAGC	65.9	65.1	67.4	65.4	67.3
CRF	**70.5**	**73.1**	**71.5**	**72.6**	**73.9**
LDNN	61.8	66.1	68.3	66.2	68.0
BLDNN	62.3	70.0	67.5	70.1	69.4
CLDNN	60.8	68.6	68.3	66.7	67.1
	P+F				
BAGC	66.2	62.8	67.4	69.3	67.5
CRF	**69.9**	**72.7**	**73.3**	**76.8**	**73.3**
LDNN	60.6	66.6	67.8	73.6	66.6
BLDNN	61.7	69.3	68.7	73.8	69.9
CLDNN	61.4	68.3	69.0	72.8	66.9
	P+F+S				
BAGC	66.0	63.1	63.4	64.6	66.5
CRF	**70.4**	**72.4**	**72.9**	**73.9**	**72.8**
LDNN	60.7	68.5	67.1	67.4	68.0
BLDNN	61.1	69.1	69.4	68.3	69.2
CLDNN	61.8	68.0	68.6	69.6	68.4

Table 2: Average weighted F1 scores for data augmentation pipeline, including *P*: paraphrasing, *F*: semantic slot filling, delexicalization, and surface realization, and *S*: synonym substitution. Best performance for AUGX=K (column) in **bold**; best performing $AUGX$ setting for each classifier (row) underlined.

Method	AUGX = 0	3	5	10	15
	Top-layer classification				
BAGC	77.4	78.9	80.3	79.2	78.8
CRF	**83.7**	**81.6**	**82.9**	**86.5**	**83.1**
LDNN	71.2	76.7	77.8	82.5	76.8
BLDNN	70.7	80.6	78.5	81.9	78.5
CLDNN	74.5	78.6	77.0	78.8	77.2
	Bottom-layer classification				
BAGC	55.8	54.0	55.9	56.2	57.2
CRF	**60.6**	63.0	**65.0**	**65.4**	**63.5**
LDNN	57.6	59.8	62.0	63.0	59.6
BLDNN	59.9	**63.5**	61.7	61.1	61.2
CLDNN	58.9	59.6	58.8	62.2	56.9

Table 3: Average weighted F1 scores for top-layer and bottom-layer classifiers. Best performing $AUGX$ setting for each classifier (row) underlined; best classifier for each $AUGX$ setting (column) in **bold**.

were not as effective compared to CRF, possibly because they suffer from overfitting at larger *AUGX* values (e.g., the best performance for the LSTM networks was for $AUGX$ values less than 15).

The P+F data augmentation setting, with $AUGX = 10$, gave us the highest results across the board (see column $AUGX = 10$ for *P+F*); again, this is confirmed with the sign test by comparing these results with the highest result by each classifier in *P* and *P+F+S* ($p = 0.001$). These findings indicate that paraphrasing alone is insufficient for diverse augmented data, and synonym substitution likely introduces too much noise.

Now focusing on the P+F data augmentation setting, for each classifier (row), ANOVA tests were performed on the results of the 5-cross validation, followed by post-hoc Tukey tests, to assess whether differences in performance are statistically significant. We found significant differences between $AUGX = 0$ and $AUGX = 10$ for LDNN ($p = 0.04$) and BLDNN ($p = 0.02$), and a trend towards significance for CLDNN ($p = 0.06$).

2-layered classification effective on conversational structure. One shortcoming of using a single classifier on the *CHICAGO-CRIME-VIS* data, is that dependencies between ARs are not being directly captured by the sequence classifiers (the context window size of each CAR can make requests too far apart in the sequence). Using 2-layered classification, we allow the top layer classifier to distinguish between AR or *think aloud* and the bottom layer classifier to determine the *AR type* if the top layer deemed the utterance an AR. The added advantage is that the top layer can model longer term dependencies of the adjacent utterances within a CAR while the bottom layer can do the same for ARs in subsequent CARs. Table 3 presents two different results, one for classification by the top layer and the other for classification using the bottom layer. For both layers, CRF is still the top performing classifier. With respect to the top layer, an ANOVA reveals a significant difference between the top performances, underlined in Table 3; post-hoc Tukey tests reveal a significant difference between CRF and CLDNN.

Domain-targeted word embeddings improve LSTM networks. We further study the 2-layered configuration in Table 3 since it achieved the best performance. We particularly focused on the effectiveness of word embeddings on the best performing top-layer and bottom-layer LSTM classifiers. Table 4 shows results when using either network trained word embeddings (e.g., embedding weights learned during network training) *L-WE*; or pre-trained GloVe (Pennington et al., 2014) embeddings *P-WE* (containing 6 billion total tokens trained on Wikipedia 2014 dump and Gigaword 5); or our domain-targeted word embeddings *D-WE* (see earlier description for details).

Whereas differences for BLDNN are not statistically significant, ANOVA revealed they are for LDNN ($p < 0.0001$); post-hoc Tukey tests reveal a statistically significant improvement of 7.9% over L-WE with D-WE, our domain-trained word embeddings; pre-trained GLOVE embeddings also result in a statistical significant difference from L-WE.

Final Model: 2-layered CRF. As we discussed, CRF performs better than all other models, across all settings as shown by sign tests, and in some cases in pairwise comparison as well. Additionally, from a practical point of view, training CRF is much faster even without using a GPU, as shown in Table 5.

Hence, we implemented a 2-layered CRF and results are shown in Table 6. For 2-layered CRF, we find a trend towards significance with a 8.6% improvement ($AUGX = 10$) wrt baseline ($AUGX = 0$) ($p = 0.09$). The top perfor-

Method	L-WE	P-WE	D-WE
	Top layer classification		
LDNN ($AUGX = 10$)	74.6	77.2	<u>82.5</u>
	Bottom layer classification		
BLDNN ($AUGX = 3$)	60.7	61.3	<u>63.5</u>

Table 4: Average weighted F1 scores for word embeddings. *L-WE*: network trained word embeddings; *P-WE*: pre-trained word embeddings; *D-WE*: our trained word embeddings. Best performance underlined.

Method	Processing Mode	Training Time (mm:ss)
CRF	CPU	00:06
BLDNN	CPU	26:08
BLDNN	GPU	13:05

Table 5: Average time (format: 2-digit minutes (mm):2-digit seconds (ss)) for CRF and BLDNN methods when using CPU or GPU.

	AUGX =				
Method	0	3	5	10	15
CRF-CRF	70.2	71.3	72.8	<u>78.8</u>	73.0

Table 6: Average weighted F1 scores for 2-layered CRF. Best performance underlined.

Label	F1
Actionable Request	77
Think Aloud	90

Table 7: Average label F1 scores for CRF Top-layer classification under $AUGX = 10$ setting.

mance by CRF in Table 6 (78.8 F1) is 2% higher (albeit not significantly so), than its top performance (76.8 F1) from Table 2. We further examined how well this top layer CRF performed on each label. Table 7 shows the average F1 scores for each label across the 5 folds. We observe that the classification model achieved a high score of 90 F1 on *think aloud* (which is expected since 1096/1536=71% of the labels are *think aloud*) while also performing relatively well on AR (77 F1).

7. Conclusions and Future Work

We have shown that 2-layered classification is effective in capturing the longer-term dependencies between utterances for DA classification in our domain. Another finding is that data augmentation is an effective technique for improving DA classification in a low resource setting, particularly the combination of paraphrasing and semantic slot filling. Finally, we observed that the LSTM networks, despite efforts to improve performance by supplying augmented data and domain-trained word embeddings, were not as effective compared to CRF, an indication that deep learning models, despite their success in a variety of domains, may not be appropriate in all settings.

As the data augmentation pipeline is concerned, our plan is to study effective strategies for filtering paraphrases that are less diverse, which we hope would reduce overfitting for larger data augmentation sizes. We also plan to study the effectiveness of our data augmentation pipeline on the coreference resolution task in our domain.

One limitation of the approach we presented is that it operates on pre-segmented CARs. As we noted earlier, we already have deployed a pipeline that takes individual ARs, and maps them to visualizations. To incorporate CARs, we will provide the pipeline with a sequence of utterances (initially transcribed, in a second phase spoken), until a mention of a slot from our KO is recognized, which indicates the starting boundary of a CAR. Then, we will continue to add utterances to the CAR until we identify an AR using the top layer of *CRF-CRF* (and then its type, with the bottom layer). Next, we will continue to add more utterances to the CAR as part of *conclusion* until an utterance with a slot from the KO is mentioned again, indicating the ending boundary of the CAR. We will repeat this process to segment subsequent CARs until there are no more incoming utterances.

8. Acknowledgements

This work was supported, initially, by NSF award IIS 1445751; and currently, by NSF award CNS 1625941, and by a UIC University Scholar award to Barbara Di Eugenio.

9. Bibliographical References

Ahmadvand, A., Choi, J. I., and Agichtein, E. (2019). Contextual dialogue act classification for open-domain conversational agents. In *The 42nd International ACM SIGIR*, pages 1273–1276.

Ang, J., Liu, Y., and Shriberg, E.). Automatic dialog act segmentation and classification in multiparty meetings. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1.

Aurisano, J., Kumar, A., Gonzales, A., Reda, K., Leigh, J., Di Eugenio, B., and Johnson, A. (2015). Show me data": Observational study of a conversational interface in visual data exploration. In *IEEE VIS*, volume 15, page 1.

Aurisano, J., Kumar, A., Gonzalez, A., Leigh, J., Di Eugenio, B., and Johnson, A. (2016). Articulate2: Toward a conversational interface for visual data exploration. In *IEEE VIS*, volume 16, page 1.

Benotti, L. (2009). Clarification potential of instructions. In *The SIGDIAL 2009 Conference*, pages 196–205, London, UK, September. Association for Computational Linguistics.

Berant, J. and Liang, P. (2014). Semantic parsing via paraphrasing. In *The 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1415–1425.

Brehmer, M. and Munzner, T. (2013). A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385.

Campagna, G., Xu, S., Moradshahi, M., Socher, R., and Lam, M. S. (2019). Genie: a generator of natural language semantic parsers for virtual assistant commands. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 394–410. ACM.

Chen, L., Javaid, M., Di Eugenio, B., and Žefran, M. (2015). The roles and recognition of haptic-ostensive actions in collaborative multimodal human-human dialogues. *Computer Speech & Language*, 32:201–231, Nov.

Cox, K., Grinter, R. E., Hibino, S. L., Jagadeesan, L. J., and Mantilla, D. (2001). A multi-modal natural language interface to an information visualization environment. *International Journal of Speech Technology*, 4(3-4):297–314.

Dhamdhere, K., McCurley, K. S., Nahmias, R., Sundararajan, M., and Yan, Q. (2017). Analyza: Exploring data with conversation. In *The 22nd International Conference on Intelligent User Interfaces*, IUI '17, pages 493–504.

Dong, L., Mallinson, J., Reddy, S., and Lapata, M. (2017). Learning to paraphrase for question answering. *arXiv preprint arXiv:1708.06022*.

Fader, A., Zettlemoyer, L., and Etzioni, O. (2013). Paraphrase-driven learning for open question answering. In *The 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1608–1618.

Gao, T., Dontcheva, M., Adar, E., Liu, Z., and Karahalios, K. G. (2015). Datatone: Managing ambiguity in natural language interfaces for data visualization. In *The 28th Annual ACM Symposium on User Interface Software & Technology*, pages 489–500. ACM.

Grammel, L., Tory, M., and Storey, M.-A. (2010). How information visualization novices construct visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):943–952.

Hakkani-Tür, D., Tür, G., Celikyilmaz, A., Chen, Y.-N., Gao, J., Deng, L., and Wang, Y.-Y. (2016). Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, pages 715–719.

Hoque, E., Setlur, V., Tory, M., and Dykeman, I. (2017). Applying pragmatics principles for interaction with visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):309–318.

Hou, Y., Liu, Y., Che, W., and Liu, T. (2018). Sequence-to-sequence data augmentation for dialogue language understanding. In *The 27th International Conference on Computational Linguistics*, pages 1234–1245, August.

Jia, R. and Liang, P. (2016). Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*.

Jovanovic, N., op den Akker, R., and Nijholt, A. (2006). Addressee identification in face-to-face meetings. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 169–176.

Jurafsky, D., Shriberg, E., and Biasca, D. (1997). Switchboard swbd-damsl labeling project coder's manual. *Draft 13. Technical Report 97-02*.

Katsakioris, M. M., Hastie, H., Konstas, I., and Laskov, A. (2019). Corpus of multimodal interaction for collaborative planning. In *Proceedings of the Combined Workshop on Spatial Language Understanding (SpLU) and Grounded Communication for Robotics (RoboNLP)*, pages 1–6.

Khanpour, H., Guntakandla, N., and Nielsen, R. (2016). Dialogue act classification in domain-independent conversations using a deep recurrent neural network. In *The 26th International Conference on Computational Linguistics (COLING)*, pages 2012–2021.

Kim, S. N., Cavedon, L., and Baldwin, T. (2010). Classifying dialogue acts in one-on-one live chats. In *The 2010 Conference on Empirical Methods in Natural Language Processing*, pages 862–871. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kumar, A., Aurisano, J., Di Eugenio, B., Johnson, A., Gonzalez, A., and Leigh, J. (2016). Towards a dialogue system that supports rich visualizations of data. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 304–309.

Kumar, A., Di Eugenio, B., Aurisano, J., Johnson, A., Alsaiari, A., Flowers, N., Gonzalez, A., and Leigh, J. (2017). Towards multimodal coreference resolution for exploratory data visualization dialogue: Context-based annotation and gesture identification. In *The 21st Workshop on the Semantics and Pragmatics of Dialogue (SemDial 2017–SaarDial)(August 2017)*, volume 48.

Kumar, H., Agarwal, A., Dasgupta, R., and Joshi, S. (2018). Dialogue act sequence labeling using hierarchical encoder with crf. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Kumar, A., Aurisano, J., Di Eugenio, B., and Johnson, A. (2020 (forthcoming)). Intelligent assistant for exploring data visualizations. In *Thirty-Third International Flairs Conference 2020*.

Li, X. and Roth, D. (2002). Learning question classifiers. In *The 19th International Conference on Computational Linguistics*, pages 1–7.

Manuvinakurike, R., Bui, T., Chang, W., and Georgila, K. (2018). Conversational image editing: Incremental intent identification in a new dialogue task. In *The 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 284–295.

Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., He, X., Heck, L., Tur, G., Yu, D., et al. (2015). Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *The Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Popescu-Belis, A. and Estrella, P. (2007). Generating us-

able formats for metadata and annotations in a large meeting corpus. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 93–96. Association for Computational Linguistics.

Popov, I. O., Schraefel, M., Hall, W., and Shadbolt, N. (2011). Connecting the dots: a multi-pivot approach to data exploration. In *International Semantic Web Conference*, pages 553–568.

Reda, K., Johnson, A. E., Leigh, J., and Papka, M. E. (2014). Evaluating user behavior and strategy during visual exploration. In *The Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pages 41–45. ACM.

Reithinger, N., Fedeler, D., Kumar, A., Lauer, C., Pecourt, E., and Romary, L. (2005). Miamm—a multimodal dialogue system using haptics. In *Advances in Natural Multimodal Dialogue Systems*, pages 307–332. Springer.

Schegloff, E. A. and Sacks, H. (1973). Opening up closings. *Semiotica*, 8(4):289–327.

Searle, J. R. (1975). Indirect Speech Acts. In P. Cole et al., editors, *Syntax and Semantics 3. Speech Acts*. Academic Press.

Setlur, V., Battersby, S. E., Tory, M., Gossweiler, R., and Chang, A. X. (2016). Eviza: A natural language interface for visual analysis. In *The 29th Annual Symposium on User Interface Software and Technology*, pages 365–377. ACM.

Shriberg, E., Dhillon, R., Bhagat, S., Ang, J., and Carvey, H. (2004). The icsi meeting recorder dialog act (mrda) corpus. In *The 5th SIGdial Workshop on Discourse and Dialogue*.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., van Ess-Dykema, C., and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *The 57th Annual Meeting of the AssStolckeociation for Computational Linguistics*, pages 3645–3650, July.

Sun, Y., Leigh, J., Johnson, A., and Lee, S. (2010). Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. In *International Symposium on Smart Graphics*, pages 184–195. Springer.

Tavafi, M., Mehdad, Y., Joty, S., Carenini, G., and Ng, R. (2013). Dialogue act recognition in synchronous and asynchronous conversations. In *The SIGDIAL 2013 Conference*, pages 117–121.

Van Someren, M., Barnard, Y., and Sandberg, J. (1994). *The think aloud method: a practical approach to modelling cognitive processes*. London: Academic Press.

Vu, N. T. (2016). Sequential convolutional neural networks for slot filling in spoken language understanding. *arXiv preprint arXiv:1606.07783*.

Wang, Y., Deng, L., and Acero, A. (2011). Semantic frame-based spoken language understanding. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pages 41–91.

Wieting, J., Mallinson, J., and Gimpel, K. (2017). Learning paraphrastic sentence embeddings from back-translated bitext. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Yoo, K. M., Shin, Y., and Lee, S.-g. (2018). Data augmentation for spoken language understanding via joint variational generation. *arXiv preprint arXiv:1809.02305*, 09.