Applying Feature Latent Semantic Analysis to Dialogue Act Classification

Abhinav Kumar
Department of Computer Science
University of Illinois at Chicago
akumar34@uic.edu

Mehrdad Alizadeh
Department of Computer Science
University of Illinois at Chicago
maliza2@uic.edu

Abstract

In this paper, we address the problem of dialogue act classification. More specifically, we apply the Feature Latent Semantic Analysis technique to the feature space representing our annotated corpus, causing the feature space to become compact and reduced in size. Next, we train traditional classifiers on the reduced feature space. Finally, with the idea that a reduced dimensional space will lead to reduced sparsity and noise in the data, we conduct multiple experiments to evaluate the effectiveness of this technique. Although applying the FLSA technique leads to slightly lower performance, we find interesting insights when comparing the FLSA technique and the traditional classification setting.

1 Introduction

Natural Language Processing has become popularized for its successful contributions to NLP-based technologies such as IBM Watson, Siri, and Google Voice. One of the crucial NLP components of such technologies is the dialogue system, which manages the dialogue conversation between the human and the software.

Motivated by the importance and popularity of improving NLP-based technologies, in this paper we focus on enhancing the effectiveness of dialogue systems by addressing the problem of dialogue act classification.

Dialogue act classification is a critical component of effective dialogue conversation. That is, we know during a conversation, that the hearer attempts to capture the intention of the speaker, and then process this interpretation in order to respond effectively. Similarly, a dialogue system must be able to process the conversation by utilizing dialogue act classification to help interpret the meaning behind each utterance.

However, the dialogue act classification problem is challenging because there is (1) ambiguity in the intention of the utterance (”Okay?” VS ”Okay’), (2) limited linguistic cues, and (3) no agreed-upon standard list of dialogue acts. Many promising statistical machine learning approaches have been proposed for the dialogue act classification problem that overcome these challenges by training on carefully selected features. One such solution uses the Feature Latent Semantic Analysis (FLSA) technique. This technique first reduces the dimensions of the feature space which consequently reduces the sparsity and noise as well. Finally, a traditional classifier is trained on this condensed feature vector representation of the training data. This approach has shown promise in earlier work because the classifiers are able to perform more efficiently and accurately due to the less sparse and noisy feature space.

In this paper, we analyze the effectiveness of the FLSA solution on the robotic-assisted elderly care setting. The three traditional classifiers we use are k-Nearest Neighbor (kNN), Multi-class Support Vector Machines (MC-SVM), and Maximum Entropy (MaxEnt). For data, the Find subcorpus of the *ELDERLY-AT-HOME* corpus is used. The subcorpus contains dialogue conversations between a robotic helper and an elderly person for finding and retrieving kitchen objects (pots, pans, etc.) from the residence. Each utterance in the conversation is annotated with dialogue acts and applicable non-verbal cues, such as pointing gestures (subject pointing at an object without physically contacting the object)and haptic-ostensive actions (subject physically touching or holding the object). We sometimes use the term multimodal information when referring to these nonverbal actions.

To evaluate the effectiveness of FLSA, we con-

duct the following four experiments: (1) running FLSA on different string feature representations, (2) running FLSA with various adjustments to the reduced feature space size parameter, (3) running FLSA on different combinations of features, and (4) running the traditional classifiers with FLSA as well as without FLSA. The experiments show that applying FLSA enables the traditional classifiers to run faster (due to a more compact feature space), however at the expense of slightly lower performance. In addition, we find that performance is improved when using the fully expanded representation of the string features. Finally, we note that MC-SVM performs the best when FLSA is applied while MaxEnt performs best in the traditional classification setting.

The remainder of the paper is structured as follows. We first review related works in section 2, which provides the reader a baseline understanding of the work accomplished in this area thus far. Next, in section 3, we discuss the annotated corpus *ELDERLY-AT-HOME* corpus and *Find* subcorpus that is used for our experiments. After the reader has gained an understanding of the work done so far and the data we will be using for our experiments, we are now ready to introduce the Feature Latent Semantic Analysis technique in section 4. With the introduction of the FLSA algorithm, the traditional classifiers are discussed next in section 5. With the machine learning algorithms introduced, we then explain the features used for the classification task in section 6. Finally, in section 7, we discuss the details of our four experiments and the results of these experiments. The last two sections, section 8 and Appendix, provide closing remarks and collaboration details for the project.

2 Related Works

Dialogue acts characterize the underlying intention of utterances (Austin, 1975). The task of automatic dialogue act classification has been widely studied for decades within several domains. (Serafin et al., 2003) showed that extending latent semantic analysis (LSA) with dialogue history features improves dialogue act classification performance. This approach of appending additional features to the orignal term frequencies feature is formally known as Feature Latent Semantic Analysis (FLSA). (Serafin and Di Eugenio, 2004) further extends the FLSA model by augmenting the model with dialogue game and speaker entity features. Realizing the improvement caused by appending features, (Di Eugenio et al., 2010) proposes an information gain based feature selection approach for FLSA to identify the most informative and useful features. This work additionally utilizes a kNN classifier that is trained on the reduced feature space generated by FLSA. This approach performed the best at the time.

In our paper, we will analyze the effectiveness of FLSA by applying the technique on a recent work, (Chen and Di Eugenio, 2013). This literature proposes the *ELDERLY-AT-HOME* corpus, which is annotated with multimodal information for each utterance in the dialogue. This study finds that MaxEnt outperforms other classifiers in terms of predicting dialogue acts. Our approach is to compare the MaxEnt performance to that of the FLSA technique on the multimodal *Find* subcorpos of the *ELDERLY-AT-HOME* corpus.

3 The ELDERLY-AT-HOME Corpus

This is a multimodal corpus (annotated with not just dialogue acts, but also multimodal information such as pointing gestures and haptic-ostensive actions), as introduced in (Chen and Di Eugenio, 2012). The corpus contains 20 human-human dialogues among an elderly (ELD) and a helper (HEL) to perform daily activities such as finding an object or cooking food.

3.1 Find Subcorpus

As mentioned in the *Previous Work* section, this paper will use the multimodal *Find* subcorpus. It contains dialogue conversations between a robotic helper and an elderly person with the purpose of finding and retrieving kitchen objects (pots, pans, etc.) from the residence. This subcorpus contains 137 find tasks. It is annotated for dialogue acts (Table 1) and modalities such as pointing gestures and haptic ostensive actions. The distribution of the annotations across the subcorpus are found in Table 2. Note that *Instruct* is the best represented while *Align* is the least represented in the annotated subcorpus. We conclude this section with an example find-task conversation between a helper and an elderly person, found in Figure 1. (Chen and Di Eugenio, 2013)

Figure 1: An example of a find task session (Chen and Di Eugenio, 2013)

1	ELD	And there is a spoon down there, in the second drawer? *[Point(ELD,Drawer1)]*
2	HEL	Down there?*[Point(HEL,Drawer1)]*
3	ELD	Yes.
4	HEL	This? *[Touch(HEL,Drawer1)]*
5	ELD	Uh-huh.
6	HEL	*[Open(HEL,Drawer1)]*
7	ELD	A spoon.
8	HEL	Is this the spoon? *[Takeout(HEL,spoon1)]*
9	ELD	No, the second drawer.
10	HEL	*[Close(HEL,Drawer1),Open(HEL,Drawer2)]*
11	ELD	Yes, there it is.
12	HEL	This one?*[Takeout(HEL,spoon2)]*
13	ELD	Yes, uh-huh.
14	HEL	OK.

Dialogue Act	ELD	HEL	Total	Ratio
Instruct	295	19	314	20.7%
Acknowledge	22	186	208	13.7%
Reply-y	179	3	182	12.0%
Check	1	155	156	10.3%
Query-yn	23	133	156	10.3%
Query-w	3	144	147	9.7%
Reply-w	132	4	136	9.0%
State-y	40	36	76	5.0%
State-n	16	50	66	4.4%
Reply-n	27	9	36	2.4%
State	7	15	22	1.5%
Explain	10	4	14	0.9%
Align	1	2	3	0.3%
Total	756	760	1516	100%

Table 1: Dialogue act counts in Find task subcorpus.

4 Feature latent semantic analysis

This section first explains the Latent Semantic Analysis (LSA) method. We then describe Feature Latent Semantic Analysis (FLSA), which extends LSA.

4.1 Latent semantic analysis

Latent Semantic Analysis (LSA) is a method for extracting (representing) the contextual-usage meaning of words by statistical computations applied to a set of text documents (Landauer et al., 1998). This technique operates on the terms of the documents with the purpose of reducing sparse linguistic data space. This leads to improved performance by preserving the most significant associations among the features.

In a vector-space model, documents are represented as vectors of term frequencies. In this way, a set of documents can be described as a term-document matrix. The LSA technique first applies singular valued decomposition (SVD) on a term-document matrix W. SVD decomposes W into three matrices, $W = USV^T$. Next, LSA reduces the dimensional space of each matrix by preserving the k strongest hidden features ($W_k = U_k S_k V_k^T$). Here k is the reduced feature space size parameter. Finally, an arbitrary document Q is transformed to the new feature space ($Q_k = Q^T U_k S_k^{-1}$).

	Elderly	Helper	Total
Utterance	756	760	1516
Words	3612	2981	6593
Pointing	219	113	332
H-O Actions	15	582	597

Table 2: Counts of utterances, words, pointing gestures, and H-O actions.

Figure 2: Term-Document matrix

	d_1	d_2	d_3	d_4	d_5
romeo	1	0	1	0	0
juliet	1	1	0	0	0
happy	0	1	0	0	0
dagger	0	1	1	0	0
live	0	0	0	1	0
die	0	0	1	1	0
free	0	0	0	1	0
new-hampshire	0	0	0	1	1

An example from (Thomo, 2009) provides a better intuition to LSA. Table 3 shows a set of 5 documents. As mentioned before, documents can be represented in vector space model. Figure 2 illustrates term frequency matrix of the documents.

id	text content
d_1	*Romeo and Juliet.*
d_2	*Juliet: O happy dagger!*
d_3	*Romeo died by dagger.*
d_4	*"Live free or die", that's the New-Hampshires motto.*
d_5	*Did you know, New-Hampshire is in New-England.*

Table 3: A sample of documents

SVD decomposes the matrix into three matrices:

$$U = \begin{bmatrix} -0.39 & 0.28 & -0.57 & 0.45 & -0.10 \\ -0.31 & 0.45 & 0.41 & 0.51 & 0.20 \\ -0.17 & 0.26 & 0.49 & -0.25 & 0.04 \\ -0.43 & 0.36 & 0.01 & -0.57 & -0.22 \\ -0.26 & -0.34 & 0.14 & 0.04 & 0.41 \\ -0.52 & -0.24 & -0.33 & -0.27 & 0.15 \\ -0.26 & -0.34 & 0.14 & 0.04 & 0.41 \\ -0.32 & -0.46 & 0.31 & 0.23 & -0.72 \end{bmatrix}$$

$$S = \begin{bmatrix} 2.285 & 0 & 0 & 0 & 0 \\ 0 & 2.010 & 0 & 0 & 0 \\ 0 & 0 & 1.361 & 0 & 0 \\ 0 & 0 & 0 & 1.118 & 0 \\ 0 & 0 & 0 & 0 & 0.797 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.31 & -0.40 & -0.59 & -0.60 & -0.14 \\ 0.36 & 0.54 & 0.20 & -0.69 & -0.22 \\ -0.11 & 0.67 & -0.69 & 0.18 & 0.23 \\ 0.86 & -0.28 & -0.35 & 0.05 & 0.21 \\ 0.12 & 0.03 & -0.20 & 0.33 & -0.91 \end{bmatrix}$$

Then, LSA sets non-significant elements of S to zero indicated by S_2:

$$S_2 = \begin{bmatrix} 2.285 & 0 \\ 0 & 2.010 \end{bmatrix}$$

In other words $k = 2$ significant semantic concepts are selected. Now $W_2 = U_2 S_2 V_2^T$ represents documents in reduced 2D feature space. The documents are represented in new vector space as follow:

$$d_1 = \begin{bmatrix} 0.711 \\ 0.730 \end{bmatrix}, d_2 = \begin{bmatrix} 0.930 \\ 1.087 \end{bmatrix}, d_3 = \begin{bmatrix} 1.357 \\ 0.402 \end{bmatrix}$$

$$, d_4 = \begin{bmatrix} 1.378 \\ -1.397 \end{bmatrix}, d_5 = \begin{bmatrix} 0.327 \\ -0.460 \end{bmatrix}$$

Now, representation of a new document $d_6 =$ *"die, dagger"* is computed below:

$$Q_2 = Q^T U_2 S_2^{-1} = \begin{bmatrix} 1.099 \\ 0.124 \end{bmatrix}$$

.

4.2 Feature Latent Semantic Analysis

The problem with LSA is that the document representation is limited to terms. In other words, an utterance has other features such as non-verbal features that seems useful if encoded. FLSA augments the term-utterance matrix with new features as virtual terms. Figure 3 shows an example of a term-utterance matrix augmented with two expanded features to represent speaker entity. FLSA matrix computations are similar to LSA. Also note that k is the reduced feature space size parameter, and we will analyze its effect on FLSA in the *Experiments and Results* section.

Figure 3: Feature-Utterance matrix(Di Eugenio et al., 2010)

	U1	U2	U3	U4	U5	U6	U7
right	1	1	0	0	0	0	0
you	0	3	0	0	0	0	1
go	0	3	0	0	0	0	1
going	0	0	2	0	0	0	0
below	0	0	0	2	0	1	0
south	0	1	0	0	0	0	0
north	0	1	0	0	0	0	0
left-hand	0	1	0	0	0	0	1
past	0	1	1	0	0	0	1
<Helper>	1	1	0	1	1	0	1
<Elderly>	0	0	1	0	0	1	0

5 Classifiers

The traditional classifiers that we will use for the actual dialogue act classification include k-Nearest Neighbor (kNN), Multi-class Support Vector Machines (MC-SVM), and Maximum Entropy (MaxEnt). Some further details are specified below.

5.1 k-Nearest Neighbor

This algorithm uses cosine similarity to calculate the similarity between the feature vector representation of each data point in the corpus and the feature vector for which prediction is being done. The k value indicates that the algorithm will select the dialogue act that is the majority representative of the k nearest neighbors. Through experimentation, we set k = 10 to ensure the best performance from this classifier.

5.2 Multi-class Support Vector Machines (MC-SVM)

Support Vector Machines (SVM) is traditionally a binary classifier. For our experiments, we apply the one-versus-one strategy in order to utilize SVM as a multi-class model. Since there are 13 dialogue act labels, our strategy involves creating $\binom{13}{2}$ pairs of binary classifiers, covering all possible combinations, and selecting the dialogue label that is selected by the most number of classifiers. One other point to note is that we experimented

with both a linear kernal and Gaussian RBF kernal, but found that MC-SVM performed best with the linear kernal.

5.3 Maximum Entropy (MaxEnt)

The MaxEnt classifier is chosen in this paper because it has been shown to perform the best in previous work (in the traditional classifier setting, that is without FLSA being applied). Hence, we want to compare the experimental results with that of MaxEnt. For this classifier, each feature adds a constraint on our total probability distribution, such that the constrained distribution approximates the distribution we see in our training data. We then choose the dialogue act label corresponding to the maximum entropy distribution that satisfies the constraints. It should also be noted that for this project, we applied L1 regularization to combat over-fitting of the data.

6 Features

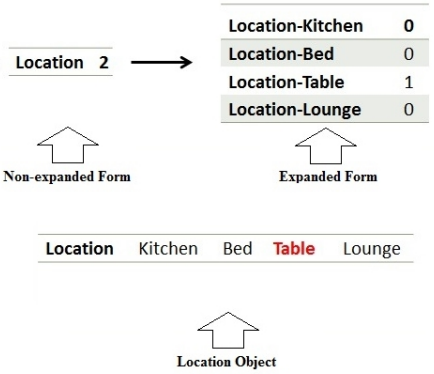
We categorize the features into five general groups: Textual (T), Multimodal (M), Utterance (U), Dialogue History (H) and Dialogue Games (G).

- *Textual Features*: From a total of 2,917 fully represented features (including the expanded string features), 98.4% of them are the textual features. In other words, the textual features dominate the other features. This category is composed of six subcategories: Word (W), Length (L), Heuristics (R), POS (P), Chuck (C), Syntax (S) and Dependency (D). Each subcategory is described below:
  - *Word*: unigrams
  - *Length*: total number of sentences and words in the utterance
  - *Heuristics*: Indicator referring to whether the utterance contains a WH word (e.g. what), or a yes/no word (e.g. yeah).
  - *POS*: Part of speech tag for each word in the utterance
  - *Chunk*: The high level parsing of an utterance
  - *Syntax*: The parse tree of the last sentence in an utterance
  - *Dependency*: The dependency parse tree of the last sentence in an utterance
- *Utterance Features*: These features are representative of the meta information of an utterance, such as the actor (elderly or helper), time in seconds of the utterance and the distance to the utterance from the beginning of the dialogue.
- *Multimodal Features*: These features include pointing gesture, haptic ostensive action, and location. The pointing gesture feature specifies whether the actor points to an object. The haptic-ostensive action feature indicates if the actor performed an action on an object. Finally, the location features represent the location of the two subject(the possible locations are kitchen, table, lounge, and bed).
- *Dialogue History Features*: These features refer to the events that occurred prior to the current utterance. The events are called *moves* in this case, and are defined as any mixture of related utterances, pointing gestures and haptic ostensive actions, made by the same subject. The features include the previous moves actor, whether the last two moves have the same actor, the previous dialogue act label, and the previous haptic-ostensive action.
- *Dialogue Game Feature*: This feature models the hierarchical dialogue structure. A dialogue game begins when the elderly person requests the helper to find a new kitchen object. If an utterance is annotated with *instruct*, *explain*, *check*, *align*, *query-yn*,or *query-w*, then this is a new dialogue game. Of course the game ends once the desired object has been found.

7 Experiments and Results

We conduct four experiments to help us assess the effectiveness of FLSA. The first experiment, running FLSA on different string feature representations, will help us select a feature representation for the term-utterance matrix that leads to improved classifier performance. The next experiment, running FLSA with varying values of the reduced space size parameter k, enables us to select the best k for FLSA. With the optimal parameters and setup from the first two experiments, we now focus on the effectiveness of FLSA. The next experiment, running FLSA on different combinations of features, provides us with insight into

Figure 4: Feature representation forms



which features are useful for the classification task. The final experiment, running the traditional classifiers with FLSA as well as without FLSA, captures how effective the traditional classifiers are when FLSA is applied.

For the results of the experiments, we ran 5-fold cross validation to compute the average accuracy and F1-score values.

7.1 Feature Representations

- *Description*: The term-utterance matrix used by FLSA expects numerical feature values and therefore we must transform the string features to numerical values. We can represent these string features in two possible ways: expanded or non-expanded form. The non-expanded form is the classical form, in which a numerical code maps to a specific string value. However, in the expanded form, only values of "0" and "1" are allowed, such that a value of "1" indicates that the string exists in the document while a "0" implies the string is non-existent in that document.

  An example of each form is illustrated in Figure 4. Here, the location of the target object is on the table. The non-expanded form represents this string feature value as "2" while in expanded form, the string feature value is "0010."

- *Results*: As we see in Figure 5, for both the FLSA and traditional classifier setting, the accuracy is highest when the expanded form of the string features is used. This is likely because of the normalizing effect found in the expanded form. That is, all of the features are between "0" and "1" and therefore the classifier is not biased towards features with values greater than "1".

Figure 5: Accuracy of expanded vs nonexpanded feature representation

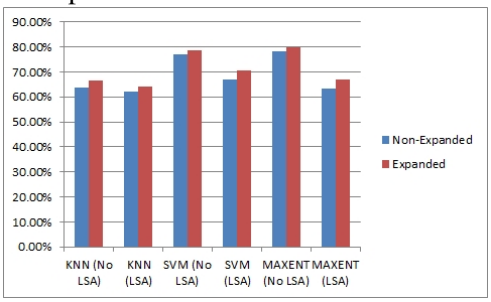
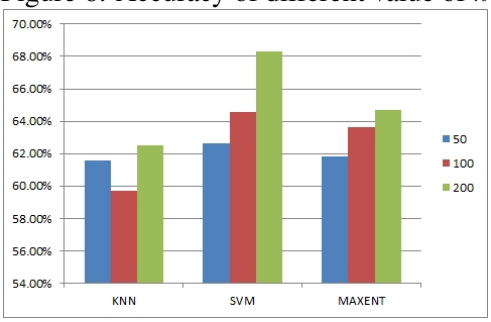


Figure 6: Accuracy of different value of k



7.2 Reduced Space Size Parameter

- *Description*: For this experiment, the original feature space is 2,917 (with the expanded string features form). The FLSA technique will reduce this feature space to k = 50, 100, and 200.

- *Results*: As we see in Figure 6, for $k = 200$, the FLSA technique had the highest accuracy across all three traditional classifiers. This is likely because some of the features are probably highly correlated, and therefore, for smaller values of k, the smaller feature space is unable to retain all of this information successfully.

  An additional observation is that the three classifiers performed slightly worse when FLSA is applied as compared to the traditional classifier setting. Again, this is likely because there is a high correlation among the textual features, so compacting the feature space size is causing too much information loss.

7.3 Feature Combinations

- *Description*: Since 98.4% of the features are textual, we focused on running FLSA on this

Figure 7: Accuracy of different set of features

Description	Features	KNN		SVM		MAXENT	
		No LSA	LSA	No LSA	LSA	No LSA	LSA
Words (W)	W	46.14%	43.50%	60.03%	50.17%	59.76%	47.33%
Length (L)	W+L	46.49%	45.74%	60.78%	49.84%	60.68%	46.67%
Wh-word (R)	W+L+R	53.62%	53.55%	65.68%	60.15%	65.79%	55.73%
POS (P)	W+L+R+P	57.84%	58.17%	65.58%	64.58%	65.67%	64.51%
Chunk (C)	W+L+R+P+C	57.01%	59.56%	65.48%	64.98%	65.51%	64.52%
Syntax (S)	W+L+R+P+C+S	58.56%	55.52%	65.72%	66.03%	66.27%	63.72%
Dependency (D)	T=W+L+R+P+C+S+D	52.17%	55.39%	64.94%	58.83%	66.31%	55.92%
Multimodal (M)	T+M	51.89%	50.04%	67.64%	57.11%	67.91%	54.93%
Utterance (U)	T+M+U	55.86%	59.36%	71.48%	64.25%	71.91%	61.47%
Dialogue History (H)	T+M+U+H	64.56%	63.33%	**79.60%**	68.28%	79.21%	63.99%
Dialogue Game (G)	T+M+U+H+G	**66.54%**	**64.58%**	78.66%	**71.13%**	**80.00%**	**68.08%**

features category. The combinations are created by incrementally augmenting a textual feature to the term-utterance matrix on each FLSA run.

- *Results*: In Figure 7, we see the outcome of this experiment. Each row represents the performance of an incrementally augmented feature combination on kNN, MC-SVM, and MaxEnt for both the FLSA and traditional classifier setting.

  We find that adding the *Wh-word* feature improves the performance across all the classifiers by 5% to 8%. This is reasonable because the question-related dialogue acts (*Query-w* and *Query-yn*) represent 20% of the dialogue act annotations in the corpus, and hence a feature that identifies wh-words will be very useful in improving prediction.

  The *Dependency* feature has a degrading affect on the classifiers, which could be explained by the fact that it is difficult to learn the dependency tree structure and hence, eliminating this feature will improve the performance.

  The *Dialogue History* and *Dialogue Games* features clearly enhance the performance. There is a correlation among the utterances of a conversation and hence, historical information (using *Dialogue History*) will help improve classification. It has also been shown that hierarchial dialogue structure (using *Dialogue Games*) also helps with classification.

7.4 Traditional VS FLSA-Based Classification

- *Description*: In this experiment, we run the kNN, MC-SVM, and MaxEnt classifiers both with and without FLSA. We also keep track of the performance in terms of each dialogue act label to obtain additional insights about individual dialogue acts.

Figure 8: F1 score of different dialogue acts using different classifier

Dialogue Act	Numbers	KNN		SVM		MAXENT	
		No LSA	LSA	No LSA	LSA	No LSA	LSA
Acknowledge	208	0.79	0.71	0.84	**0.78**	**0.88**	**0.78**
Align	3	0.00	0.00	0.00	0.00	0.00	0.00
Check	156	0.82	0.80	0.90	**0.90**	**0.96**	0.88
Explain	14	0.00	0.00	0.00	0.00	0.00	0.00
Instruct	314	0.83	0.76	0.89	**0.81**	**0.91**	0.80
Query-w	147	0.44	0.52	0.87	**0.67**	**0.88**	0.65
Query-yn	156	0.65	0.56	0.75	**0.70**	**0.79**	0.68
Reply-n	36	0.67	0.70	**0.95**	**0.88**	**0.95**	0.71
Reply-w	136	0.76	0.70	0.78	0.60	**0.86**	**0.68**
Reply-y	182	0.86	0.85	0.92	**0.92**	**0.95**	0.88
State	22	0.00	0.00	0.00	0.00	0.00	0.00
State-n	66	0.82	0.77	**0.96**	**0.85**	0.94	0.77
State-y	76	0.43	0.42	**0.83**	**0.50**	**0.83**	0.40
Average		0.76	0. 70	0.87	**0.78**	**0.90**	0.76

- *Results*: In Figure 8, we find several interesting insights. The distribution of the dialogue act annotations across the corpus is found under the *Numbers* column. The class inbalance found here (for example, *Instruct* annotation occurs twice as often as *Query-yn*) plays a significant role in performance. While *Instruct*, which is the best represented, corresponds to high F1-scores across the classifiers, the least represented dialogue acts *Align*, *Explain*, and *State* correspond to the worst performance. This is expected since the classifiers will have far more training instances of *Instruct* in comparison to *Align*, *Explain*, and *State*. With a lack of enough training instances for some of the dialogue act labels, the classifiers will not be able to predict those labels successfully.

  The final observation we note is that MC-SVM performs the best when FLSA is applied while MaxEnt performs best in the traditional classifier setting. We believe that MC-SVM performs best when FLSA is applied because the lower dimensional feature space removes a significant amount of noise and hence the hyperplane can be better drawn for improved classification.

8 Conclusion and future work

In this paper, we applied the Feature Latent Semantic Analysis (FLSA) technique on the multimodal *Find* subcorpus in the robotic-assisted elderly care setting. To understand the effectiveness of the FLSA technique on the multimodal corpus, we conducted four experiments. From the experiments, we found that the FLSA technique performed slighly worse than the traditional classifier setting. We attribute the degradation in performance to the high correlation and sensitivity of

the features, hence compacting the feature space results in the classifiers unable to learn the underlying correlation in the features.

In the future, we plan to conduct another experiment in which we explore how well FLSA performs if we eliminate the *Dependency* feature, since we saw in our experiments, that it seems to degrade the overall performance. We also would like to try additional classifiers for the FLSA technique, including Naive Bayes and Decision Tree Classifier to observe if FLSA may possibly improve the performance using those additional machine learning algorithms.

Appendix

For this project, we initially analyzed the code and corpus provided by Lin Chen together to have a better idea of how his code is able to read from the *Find* subcorpus. In the initial stages, we also explored various machine learning libraries in both Java and Python to identify a library for Latent Semantic Analysis that can handle large amounts of data quickly and with minimal memory use. We found Weka to be far too slow and unable to scale to the large size that is needed. We finally chose the Gensim library in Python because it is extremely fast for Latent Semantic Analysis.

After the initial setup, the work was separated into two major tasks. Mehrdad focused on reading the data from the annotated corpus, extract the desired features, and finally normalize the features to the expanded features style. In addition, Mehrdad was also deeply invovled with the contents of (Di Eugenio et al., 2010) and (Chen and Di Eugenio, 2013) to assist in implementing the FLSA technique. Abhinav focused on implementing the FLSA technique using the Gensim library, as well as using the Sklearn library in Python to compute 5-fold cross validation so that accuracy and F1-scores could be computed, and also added KNN, MC-SVM, and MaxEnt models using libraries from Sklearn.

Finally, for the documentation and presentation tasks, we collaborated together to complete this work. In the end, our responsibilites and contributions are equal on the project. We are very excited that the project is able to successfully perform classification with reasonable performance results.

References

John Langshaw Austin. 1975. *How to do things with words*, volume 1955. Oxford university press.

Lin Chen and Barbara Di Eugenio. 2012. Co-reference via pointing and haptics in multi-modal dialogues. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 523–527. Association for Computational Linguistics.

Lin Chen and Barbara Di Eugenio. 2013. Multimodality and dialogue act classification in the robohelper project. pages 183–192.

Barbara Di Eugenio, Zhuli Xie, and Riccardo Serafin. 2010. Dialogue act classification, higher order dialogue structure, and instance-based learning. *Dialogue & Discourse*, 1(2):1–24.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

Riccardo Serafin and Barbara Di Eugenio. 2004. Flsa: Extending latent semantic analysis with features for dialogue act classification. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 692. Association for Computational Linguistics.

Riccardo Serafin, Barbara Di Eugenio, and Michael Glass. 2003. Latent semantic analysis for dialogue act classification. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers-Volume 2*, pages 94–96. Association for Computational Linguistics.

Alex Thomo. 2009. Latent semantic analysis tutorial.