FLSA: Extending Latent Semantic Analysis with features for dialogue act classification

Riccardo Serafin
CEFRIEL
Via Fucini 2
20133 Milano, Italy
Riccardo.Serafin@students.cefriel.it

Barbara Di Eugenio
Computer Science
University of Illinois
Chicago, IL 60607 USA
bdieugen@cs.uic.edu

Abstract

We discuss Feature Latent Semantic Analysis (FLSA), an extension to Latent Semantic Analysis (LSA). LSA is a statistical method that is ordinarily trained on words only; FLSA adds to LSA the richness of the many other linguistic features that a corpus may be labeled with. We applied FLSA to dialogue act classification with excellent results. We report results on three corpora: CallHome Spanish, MapTask, and our own corpus of tutoring dialogues.

1 Introduction

In this paper, we propose Feature Latent Semantic Analysis (FLSA) as an extension to Latent Semantic Analysis (LSA). LSA can be thought as representing *the meaning of a word as a kind of average of the meanings of all the passages in which it appears, and the meaning of a passage as a kind of average of the meaning of all the words it contains* (Landauer and Dumais, 1997). It builds a semantic space where words and passages are represented as vectors. LSA is based on Single Value Decomposition (SVD), a mathematical technique that *causes the semantic space to be arranged so as to reflect the major associative patterns in the data*. LSA has been successfully applied to many tasks, such as assessing the quality of student essays (Foltz et al., 1999) or interpreting the student's input in an Intelligent Tutoring system (Wiemer-Hastings, 2001).

A common criticism of LSA is that it uses only words and ignores anything else, e.g. syntactic information: to LSA, *man bites dog* is identical to *dog bites man*. We suggest that an LSA semantic space can be built from the co-occurrence of arbitrary textual features, not just words. We are calling LSA augmented with features FLSA, for Feature LSA. Relevant prior work on LSA only includes Structured Latent Semantic Analysis (Wiemer-Hastings, 2001), and the predication algorithm of (Kintsch, 2001). We will show that for our task, dialogue act classification, syntactic features do not help, but most dialogue related features do. Surprisingly, one dialogue related feature that does not help is the dialogue act history.

We applied LSA / FLSA to dialogue act classification. Dialogue systems need to perform dialogue act classification, in order to understand the role the user's utterance plays in the dialogue (e.g., a question for information or a request to perform an action). In recent years, a variety of empirical techniques have been used to train the dialogue act classifier (Samuel et al., 1998; Stolcke et al., 2000). A second contribution of our work is to show that FLSA is successful at dialogue act classification, reaching comparable or better results than other published methods. With respect to a baseline of choosing the most frequent dialogue act (DA), LSA reduces error rates between 33% and 52%, and FLSA reduces error rates between 60% and 78%.

LSA is an attractive method for this task because it is straightforward to train and use. More importantly, although it is a statistical theory, it has been shown to mimic many aspects of human competence / performance (Landauer and Dumais, 1997). Thus, it appears to capture important components of meaning. Our results suggest that LSA / FLSA do so also as concerns DA classification. On MapTask, our FLSA classifier agrees with human coders to a satisfactory degree, and makes most of the same mistakes.

2 Feature Latent Semantic Analysis

We will start by discussing LSA. The input to LSA is a Word-Document matrix W with a row for each word, and a column for each *document* (for us, a document is a unit, e.g. an utterance, tagged with a DA). Cell $c(i, j)$ contains the frequency with which $word_i$ appears in $document_j$.[1] Clearly, this $w \times d$ matrix W will be very sparse. Next, LSA applies

[1] Word frequencies are normally weighted according to specific functions, but we used raw frequencies because we wanted to assess our extensions to LSA independently from any bias introduced by the specific weighting technique.

to W Singular Value Decomposition (SVD), to decompose it into the product of three other matrices, $W = T_0 S_0 D_0^T$, so that T_0 and D_0 have orthonormal columns and S_0 is diagonal. SVD then provides a simple strategy for optimal approximate fit using smaller matrices. If the singular values in S_0 are ordered by size, the first k largest may be kept and the remaining smaller ones set to zero. The product of the resulting matrices is a matrix $\hat{W}$ of rank k which is approximately equal to W; it is the matrix of rank k with the best possible least-squares-fit to W.

The number of dimensions k retained by LSA is an empirical question. However, crucially k is much smaller than the dimension of the original space. The results we will report later are for the best k we experimented with.

Figure 1 shows a hypothetical dialogue annotated with MapTask style DAs. Table 1 shows the Word-Document matrix W that LSA starts with – note that as usual stop words such as *a, the, you* have been eliminated. [2] Table 2 shows the approximate representation of W in a much smaller space.

To choose the best tag for a document in the test set, we first compute its vector representation in the semantic space LSA computed, then we compare the vector representing the new document with the vector of each document in the training set. The tag of the document which has the highest similarity with our test vector is assigned to the new document – it is customary to use the cosine between the two vectors as a measure of similarity. In our case, the new document is a unit (utterance) to be tagged with a DA, and we assign to it the DA of the document in the training set to which the new document is most similar.

Feature LSA. In general, in FLSA we add extra features to LSA by adding a new "word" for each value that the feature of interest can take (in some cases, e.g. when adding POS tags, we extend the matrix in a different way — see Sec. 4). The only assumption is that there are one or more non word related features associated with each document that can take a finite number of values. In the Word–Document matrix, the word index is increased to include a new place holder for each possible value the feature may take. When creating the matrix, a count of one is placed in the rows related to the new indexes if a particular feature applies to the document under analysis. For instance, if we wish to include the speaker identity as a new feature for the dialogue in Figure 1, the initial Word–Document matrix will be modified as in Table 3 (its first 14 rows are as in Table 1).

This process is easily extended if more than one non-word feature is desired per document, if more than one feature value applies to a single document or if a single feature appears more than once in a document (Serafin, 2003).

3 Corpora

We report experiments on three corpora, Spanish CallHome, MapTask, and DIAG-NLP.

The Spanish CallHome corpus (Levin et al., 1998; Ries, 1999) comprises 120 unrestricted phone calls in Spanish between family members and friends, for a total of 12066 unique words and 44628 DAs. The Spanish CallHome corpus is annotated at three levels: DAs, dialogue games and dialogue activities. The DA annotation augments a basic tag such as *statement* along several dimensions, such as whether the statement describes a psychological state of the speaker. This results in 232 different DA tags, many with very low frequencies. In this sort of situations, tag categories are often collapsed when running experiments so as to get meaningful frequencies (Stolcke et al., 2000). In CallHome37, we collapsed different types of statements and backchannels, obtaining 37 different tags. CallHome37 maintains some subcategorizations, e.g. whether a question is yes/no or rhetorical. In CallHome10, we further collapse these categories. CallHome10 is reduced to 8 DAs proper (e.g., statement, question, answer) plus the two tags ''%'' for abandoned sentences and ''x'' for noise.
CallHome Spanish is further annotated for dialogue games and activities. Dialogue game annotation is based on the MapTask notion of a dialogue game, *a set of utterances starting with an initiation and encompassing all utterances up until the purpose of the game has been fulfilled (e.g., the requested information has been transferred) or abandoned* (Carletta et al., 1997). Moves are the components of games, they correspond to a single or more DAs, and each is tagged as Initiative, Response or Feedback. Each game is also given a label, such as *Info(rmation)* or *Direct(ive)*. Finally, activities pertain to the main goal of a certain discourse stretch, such as *gossip* or *argue*.

The HCRC MapTask corpus is a collection of dialogues regarding a "Map Task" experiment. Two participants sit opposite one another and each of them receives a map, but the two maps differ. The *instruction giver (G)*'s map has a route indicated while *instruction follower (F)*'s map does not in-

[2] We use a very short list of stop words (< 50), as our experiments revealed that for dialogue act annotation LSA is sensitive to the most common words too. This is why *to* is included in Table 1.

(Doc 1) *G: Do you see the lake with the black swan?*	*Query–yn*
(Doc 2) *F: Yes, I do*	*Reply–y*
(Doc 3) *G: Ok,*	*Ready*
(Doc 4) *G: draw a line straight to it*	*Instruct*
(Doc 5) *F: straight to the lake?*	*Check*
(Doc 6) *G: yes, that's right*	*Reply–y*
(Doc 7) *F: Ok, I'll do it*	*Acknowledge*

Figure 1: A hypothetical dialogue annotated with MapTask tags

	(Doc 1)	(Doc 2)	(Doc 3)	(Doc 4)	(Doc 5)	(Doc 6)	(Doc 7)
do	1	1	0	0	0	0	1
see	1	0	0	0	0	0	0
lake	1	0	0	0	1	0	0
black	1	0	0	0	0	0	0
swan	1	0	0	0	0	0	0
yes	0	1	0	0	0	1	0
ok	0	0	1	0	0	0	1
draw	0	0	0	1	0	0	0
line	0	0	0	1	0	0	0
straight	0	0	0	1	1	0	0
to	0	0	0	1	1	0	0
it	0	0	0	1	0	0	1
that	0	0	0	0	0	1	0
right	0	0	0	0	0	1	0

Table 1: The 14-dimensional word-document matrix W

clude the drawing of the route. The task is for G to give directions to F, so that, at the end, F is able to reproduce G's route on her map. The MapTask corpus is composed of 128 dialogues, for a total of 1,835 unique words and 27,084 DAs. It has been tagged at various levels, from POS to disfluencies, from syntax to DAs. The MapTask coding scheme uses 13 DAs (called moves), that include: *Instruct* (a request that the partner carry out an action), *Explain* (one of the partners states some information that was not explicitly elicited by the other), *Query-yn/-w*, *Acknowledge*, *Reply-y/-n/-w* and others. The MapTask corpus is also tagged for games as defined above, but differently from CallHome, 6 DAs are identified as potential initiators of games (of course not every initiator DA initiates a game). Finally, transactions provide the subdialogue structure of a dialogue; each is built of several dialogue games and corresponds to one step of the task.

DIAG-NLP is a corpus of computer mediated tutoring dialogues between a tutor and a student who is diagnosing a fault in a mechanical system with a tutoring system built with the DIAG authoring tool (Towne, 1997). The student's input is via menu, the tutor is in a different room and answers via a text window. The DIAG-NLP corpus comprises 23 'dialogues' for a total of 607 unique words and 660 DAs (it is thus much smaller than the other two). It has been annotated for a variety of features, including four DAs[3] (Glass et al., 2002): *problem solving*, the tutor gives problem solving directions; *judgment*, the tutor evaluates the student's actions or diagnosis; *domain knowledge*, the tutor imparts domain knowledge; and *other*, when none of the previous three applies. Other features encode domain objects and their properties, and *Consult Type*, the type of student query.

4 Results

Table 4 reports the results we obtained for each corpus and method (to train and evaluate each method, we used 5-fold cross-validation). We include the baseline, computed as picking the most frequent DA

[3]They should be more appropriately termed *tutor moves*.

	(Doc 1)	(Doc 2)	(Doc 3)	(Doc 4)	(Doc 5)	(Doc 6)	(Doc 7)
Dim. 1	1.3076	0.4717	0.1529	1.6668	1.1737	0.1193	0.9101
Dim. 2	1.5991	0.6797	0.0958	-1.3697	-0.4771	0.2844	0.4205

Table 2: The reduced 2-dimensional matrix $\hat{W}$

	(Doc 1)	(Doc 2)	(Doc 3)	(Doc 4)	(Doc 5)	(Doc 6)	(Doc 7)
do	1	1	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
right	0	0	0	0	0	1	0
<Giver>	1	0	1	1	0	1	0
<Follower>	0	1	0	0	1	0	1

Table 3: Word-document matrix W augmented with speaker identity

in each corpus;[4] the accuracy for LSA; the best accuracy for FLSA, and with what combination of features it was obtained; the best published result, taken from (Ries, 1999) and from (Lager and Zinovjeva, 1999) respectively for CallHome and for MapTask. Finally, for both LSA and FLSA, Table 4 includes, in parenthesis, the dimension k of the reduced semantic space. For each LSA method and corpus, we experimented with values of k between 25 and 350. The values of k that give us the best resuls for each method were thus selected empirically.

In all cases, we can see that LSA performs much better than baseline. Moreover, we can see that FLSA further improves performance, dramatically in the case of MapTask. FLSA reduces error rates between 60% and 78%, for all corpora other than DIAG-NLP (all differences in performance between LSA and FLSA are significant, other than for DIAG-NLP). DIAG-NLP may be too small a corpus to train FLSA; or *Consult Type* may not be effective, but it was the only feature appropriate for FLSA (Sec. 5 discusses how we chose appropriate features). Another extension to LSA we developed, Clustered LSA, did give an improvement in performance for DIAG (79.24%) — please see (Serafin, 2003).

As regards comparable approaches, the performance of FLSA is as good or better. For Spanish CallHome, (Ries, 1999) reports 76.2% accuracy with a hybrid approach that couples Neural Networks and ngram backoff modeling; the former uses prosodic features and POS tags, and interestingly works best with unigram backoff modeling, i.e., without taking into account the DA history – see our discussion of the ineffectiveness of the DA history below. However, (Ries, 1999) does not mention his target classification, and the reported baseline of picking the most frequent DA appears compatible with both CallHome37 and CallHome10.[5] Thus, our results with FLSA are slightly worse (- 1.33%) or better (+ 2.68%) than Ries', depending on the target classification. On MapTask, (Lager and Zinovjeva, 1999) achieves 62.1% with Transformation Based Learning using single words, bigrams, word position within the utterance, previous DA, speaker and change of speaker. We achieve much better performance on MapTask with a number of our FLSA models.

As regards results on DA classification for other corpora, the best performances obtained are up to 75% for task-oriented dialogues such as Verbmobil (Samuel et al., 1998). (Stolcke et al., 2000) reports an impressive 71% accuracy on transcribed Switchboard dialogues, using a tag set of 42 DAs. These are unrestricted English telephone conversations between two strangers that discuss a general interest topic. The DA classification task appears more difficult for corpora such as Switchboard and CallHome Spanish, that cannot benefit from the regularities imposed on the dialogue by a specific task. (Stolcke et al., 2000) employs a combination of HMM, neural networks and decision trees trained on all available features (words, prosody, sequence of DAs and speaker identity).

Table 5 reports a breakdown of the experimental results obtained with FLSA for the three tasks for which it was successful (Table 5 does not include k, which is always 25 for CallHome37 and CallHome10, and varies between 25 and 75 for MapTask). For each corpus, under the line we find results that are significantly better than those obtained with LSA. For MapTask, the first 4 results that are

[4]The baselines for CallHome37 and CallHome10 are the same because in both *statement* is the most frequent DA.

[5]An inquiry to clarify this issue went unanswered.

Corpus	Baseline	LSA	FLSA	Features	Best known result
CallHome37	42.68%	65.36% ($k = 50$)	74.87% ($k = 25$)	Game + Initiative	76.20%
CallHome10	42.68%	68.91% ($k = 25$)	78.88% ($k = 25$)	Game + Initiative	76.20%
MapTask	20.69%	42.77% ($k = 75$)	73.91% ($k = 25$)	Game + Speaker	62.10%
DIAG-NLP	43.64%	75.73% ($k = 50$)	74.81% ($k = 50$)	Consult Type	n.a.

Table 4: Accuracy for LSA and FLSA

Corpus	accuracy	Features
CallHome37	62.58%	Previous DA
CallHome37	71.08%	Initiative
CallHome37	72.69%	Game
CallHome37	74.87%	Game+Initiative
CallHome10	68.32%	Previous DA
CallHome10	73.97%	Initiative
CallHome10	76.52%	Game
CallHome10	78.88%	Game+Initiative
MapTask	41.84%	SRule
MapTask	43.28%	POS
MapTask	43.59%	Duration
MapTask	46.91%	Speaker
MapTask	47.09%	Previous DA
MapTask	66.00%	Game
MapTask	69.37%	Game+Prev. DA
MapTask	73.25%	Game+Speaker+Prev. DA
MapTask	73.91%	Game+Speaker

Table 5: FLSA Accuracy

better than LSA (from POS to Previous DA) are still pretty low; there is a difference of 19% in performance for FLSA when *Previous DA* is added and when *Game* is added.

Analysis. A few general conclusions can be drawn from Table 5, as they apply in all three cases. First, using the previous DA does not help, either at all (CallHome37 and CallHome10), or very little (MapTask). Increasing the length of the dialogue history does not improve performance. In other experiments, we increased the length up to $n = 4$: we found that the higher n, the worse the performance. As we will see in Section 5, introducing any new feature results in a larger and sparser initial matrix, which makes the task harder for FLSA; to be effective, the amount of information provided by the new feature must be sufficient to overcome this handicap. It is clear that, the longer the dialogue history, the sparser the initial matrix becomes, which explains why performance decreases. However, this does not explain why using even only the previous DA does not help. This implies that the previous DA does not provide a lot of information, as in fact is shown numerically in Section 5. This is surprising because the DA history is usually considered an important determinant of the current DA (but (Ries, 1999) observed the same).

Second, the notion of *Game* appears to be really powerful, as it vastly improves performance on two very different corpora such as CallHome and MapTask.[6] We will come back to discussing the usage of *Game* in a real dialogue system in Section 6.

Third, the syntactic features we had access to do not seem to improve performance (they were available only for MapTask). In MapTask *SRule* indicates the main structure of the utterance, such as *Declarative* or *Wh-question*. It is not surprising that *SRule* did not help, since it is well known that syntactic form is not predictive of DAs, especially those of *indirect speech act* flavor (Searle, 1975). POS tags don't help LSA either, as has already been observed by (Wiemer-Hastings, 2001; Kanejiya et al., 2003) for other tasks. The likely reason is that it is necessary to add a different 'word' for each distinct pair *word-POS*, e.g., *route* becomes split as *route-NN* and *route-VB*. This makes the Word-Document matrix much sparser: for MapTask, the number of rows increases from 1,835 for plain LSA to 2,324 for FLSA.

These negative results on adding syntactic information to LSA may just reinforce one of the claims of the LSA proponents, that structural information is irrelevant for determining meaning (Landauer and Dumais, 1997). Alternatively, syntactic information may need to be added to LSA in different ways. (Wiemer-Hastings, 2001) discusses applying LSA to each syntactic component of the sentence (subject, verb, rest of sentence), and averaging out those three measures to obtain a final similarity measure. The results are better than with plain LSA. (Kintsch, 2001) proposes an algorithm that successfully differentiates the senses of predicates on the basis on their arguments, in which *items of the semantic neighborhood of a predicate that are relevant to an argument are combined with the [LSA] predicate vector ... through a spreading activation process*.

[6] Using *Game* in MapTask does not introduce circularity, even if a game is identified by its initiating DA. We checked the matching rates for initiating and non initiating DAs with the FLSA model which employs *Game* + *Speaker*: they are 78.12% and 71.67% respectively. Hence, even if *Game* makes initiating moves easier to classify, it is highly beneficial for the classification of non initiating moves as well.

5 How to select features for FLSA

An important issue is how to select features for FLSA. One possible answer is to exhaustively train every FLSA model that corresponds to one possible feature combination. The problem is that training LSA models is in general time consuming. For example, training each FLSA model on CallHome37 takes about 35 minutes of CPU time, and on MapTask 17 minutes, on computers with one Pentium 1.7Ghz processor and 1Gb of memory. Thus, it would be better to focus only on the most promising models, especially when the number of features is high, because of the exponential number of combinations. In this work, we trained FLSA on each individual feature. Then, we trained FLSA on each feature combinations that we expected to be effective, either because of the good performances of each individual feature, or because they include features that are deemed predictive of DAs, such as the previous DA(s), even if they did not perform well individually.

After we ran our experiments, we performed a post hoc analysis based on the notion of *Information Gain* (IG) from decision tree learning (Quinlan, 1993). One approach to choosing the next feature to add to the tree at each iteration is to pick the one with the highest IG. Suppose the data set **S** is classified using n categories $v_1 ... v_n$, each with probability p_i. **S**'s entropy H can be seen as an indicator of how uncertain the outcome of the classification is, and is given by:

$$H(\mathbf{S}) = -\sum_{i=1}^{n} p_i log_2(p_i) \qquad (1)$$

If feature F divides **S** into k subsets $\mathbf{S_1} ... \mathbf{S_k}$, then IG is the expected reduction in entropy caused by partitioning the data according to the values of F:

$$IG(\mathbf{S}, A) = H(\mathbf{S}) - \sum_{i=1}^{k} \frac{|\mathbf{S_i}|}{|\mathbf{S}|} H(\mathbf{S_i}) \qquad (2)$$

In our case, we first computed the entropy of the corpora with respect to the classification induced by the DA tags (see Table 6, which also includes the LSA accuracy for convenience). Then, we computed the IG of the features or feature combinations we used in the FLSA experiments.

Table 7 reports the IG for most of the features from Table 5; it is ordered by FLSA performance. On the whole, IG appears to be a reasonably accurate predictor of performance. When a feature or feature combination has a high IG, e.g. over 1, there is also a high performance improvement. Occasionally, if the IG is small this does not hold. For example, using the previous DA reduces the entropy by 0.21 for CallHome37, but performance actually decreases. Most likely, the amount of new information introduced is rather low and it is overcome by having a larger and sparser initial matrix, which makes the task harder for FLSA. Also, when performance improves it does not necessarily increase linearly with IG (see e.g. *Game + Speaker + Previous DA* and *Game + Speaker* for MapTask). Nevertheless, IG can be effectively used to weed out unpromising features, or to rank feature combinations so that the most promising FLSA models can be trained first.

Corpus	Entropy	LSA
CallHome37	3.004	65.36%
CallHome10	2.51	68.91%
MapTask	3.38	42.77%

Table 6: Entropy measures

Corpus	Features	IG	FLSA
CallHome37	Previous DA	0.21	62.58%
CallHome37	Initiative	0.69	71.08%
CallHome37	Game	0.59	72.69%
CallHome37	Game+Initiative	1.09	74.87%
CallHome10	Previous DA	0.13	68.32%
CallHome10	Initiative	0.53	73.97%
CallHome10	Game	0.53	76.52%
CallHome10	Game+Initiative	1.01	78.88%
MapTask	Duration	0.54	43.59%
MapTask	Speaker	0.31	46.91%
MapTask	Prev. DA	0.58	47.09%
MapTask	Game	1.21	66.00%
MapTask	Game+Speaker+Prev. DA	2.04	73.25%
MapTask	Game+Speaker	1.62	73.91%

Table 7: Information gain for FLSA

6 Discussion and future work

In this paper, we have presented a novel extension to LSA, that we have called Feature LSA. Our work is the first to show that FLSA is more effective than LSA, at least for the specific task we worked on, DA classification. In parallel, we have shown that FLSA can be effectively used to train a DA classifier. We have reached performances comparable to or better than published results on DA classification, and we have used an easily trainable method.

FLSA also highlights the effectiveness of other dialogue related features, such as *Game*, to classify DAs. The drawback of features such as *Game* is that

Corpus	FLSA
CallHome37	0.676
CallHome10	0.721
MapTask	0.740

Table 8: κ measures of agreement

a dialogue system may not have them at its disposal when doing DA classification in real time. However, this problem may be circumvented. The number of different games is in general rather low (8 in CallHome Spanish, 6 in MapTask), and the game label is constant across DAs belonging to the same game. Each DA can be classified by augmenting it with each possible game label, and by choosing the most accurate match among those returned by each of these classification attempts. Further, if the system can reliably recognize the end of a game, the method just described needs to be used only for the first DA of each game. Then, the game label that gives the best result becomes the game label used for the next few DAs, until the end of the current game is detected.

Another reason why we advocate FLSA over other approaches is that it appears to be close to human performance for DA classification, in the same way that LSA approximates well many aspects of human competence / performance (Landauer and Dumais, 1997).

To support this claim, first, we used the κ coefficient (Krippendorff, 1980; Carletta, 1996) to assess the agreement between the classification made by FLSA and the classification from the corpora — see Table 8. A general rule of thumb on how to interpret the values of κ (Krippendorff, 1980) is to require a value of $\kappa \geq 0.8$, with $0.67 < \kappa < 0.8$ allowing tentative conclusions to be drawn. As a whole, Table 8 shows that FLSA achieves a satisfying level of agreement with human coders. To put Table 8 in perspective, note that expert human coders achieved $\kappa = 0.83$ on DA classification for MapTask, but also had available the speech source (Carletta et al., 1997).

We also compared the confusion matrix from (Carletta et al., 1997) with the confusion matrix we obtained for our best result on MapTask (FLSA using *Game* + *Speaker*). For humans, the largest sources of confusion are between: *check* and *query-yn*; *instruct* and *clarify*; and *acknowledge*, *reply-y* and *ready*. Likewise, our FLSA method makes the most mistakes when distinguishing between *instruct* and *clarify*; and *acknowledge*, *reply-y*, and *ready*. Instead it performs better than humans on distinguishing *check* and *query-yn*. Thus, most of the sources of confusion for humans are the same as for FLSA.

Future work includes further investigating how to select promising feature combinations, e.g. by using logical regression.

We are also exploring whether FLSA can be used as the basis for semi-automatic annotation of dialogue acts, to be incorporated into MUP, an annotation tool we have developed (Glass and Di Eugenio, 2002). The problem is that large corpora are necessary to train methods based on LSA. This would seem to defeat the purpose of using FLSA as the basis for semi-automatic dialogue annotation, since, to train FLSA in a new domain, we would need a large hand annotated corpus to start with. *Co-training* (Blum and Mitchell, 1998) may offer a solution to this problem. In co-training, two different classifiers are initially trained on a small set of annotated data, by using different features. Afterwards, each classifier is allowed to label some unlabelled data, and picks its most confidently predicted positive and negative examples; this data is added to the annotated data. The process repeats until the desired perfomance is achieved. In our scenario, we will experiment with training two different FLSA models, or one FLSA model and a different classifier, such as a naive Bayes classifier, on a small portion of annotated data that includes features like DAs, Game, etc. We will then proceed as described on the unlabelled data.

Finally, we have started applying FLSA to a different problem, that of judging the coherence of texts. Whereas LSA has been already successfully applied to this task (Foltz et al., 1998), the issue is whether FLSA could perform better by also taking into account those features of a text that enhance its coherence for humans, such as appropriate cue words.

Acknowledgments

This work is supported by grant N00014-00-1-0640 from the Office of Naval Research, and in part, by award 0133123 from the National Science Foundation. Thanks to Michael Glass for initially suggesting extending LSA with features and to HCRC (University of Edinburgh) for sharing their annotated MapTask corpus. The work was performed while the first author was at the University of Illinois in Chicago.

References

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT98, Proceedings of the Conference on Computational Learning Theory*.

Jean Carletta, Amy Isard, Stephen Isard, Jacqueline C. Kowtko, Gwyneth Doherty-Sneddon, and Anne H. Anderson. 1997. The reliability of a dialogue structure coding scheme. *Computational Lingustics*, 23(1):13–31.

Jean Carletta. 1996. Assessing agreement on classification tasks: the Kappa statistic. *Computational Linguistics*, 22(2):249–254.

Peter W. Foltz, Walter Kintsch, and Thomas K. Landauer. 1998. The measurement of textual coherence with Latent Semantic Analysis. *Discourse Processes*, 25:285–308.

Peter W. Foltz, Darrell Laham, and Thomas K. Landauer. 1999. The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 1(2).

Michael Glass and Barbara Di Eugenio. 2002. MUP: The UIC standoff markup tool. In *The Third SigDIAL Workshop on Discourse and Dialogue*, Philadelphia, PA, July.

Michael Glass, Heena Raval, Barbara Di Eugenio, and Maarika Traat. 2002. The DIAG-NLP dialogues: coding manual. Technical Report UIC-CS 02-03, University of Illinois - Chicago.

Dharmendra Kanejiya, Arun Kumar, and Surendra Prasad. 2003. Automatic Evaluation of Students' Answers using Syntactically Enhanced LSA. In *HLT-NAACL Workshop on Building Educational Applications using Natural Language Processing*, pages 53–60, Edmonton, Canada.

Walter Kintsch. 2001. Predication. *Cognitive Science*, 25:173–202.

Klaus Krippendorff. 1980. *Content Analysis: an Introduction to its Methodology*. Sage Publications, Beverly Hills, CA.

T. Lager and N. Zinovjeva. 1999. Training a dialogue act tagger with the μ-TBL system. In *The Third Swedish Symposium on Multimodal Communication*, Linköping University Natural Language Processing Laboratory (NLPLAB).

Thomas K. Landauer and S.T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Lori Levin, Ann Thymé-Gobbel, Alon Lavie, Klaus Ries, and Klaus Zechner. 1998. A discourse coding scheme for conversational Spanish. In *Proceedings ICSLP*.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Klaus Ries. 1999. HMM and Neural Network Based Speech Act Detection. In *Proceedings of ICASSP 99*, Phoenix, Arizona, March.

Ken Samuel, Sandra Carberry, and K. Vijay-Shanker. 1998. Dialogue act tagging with transformation-based learning. In *ACL/COLING 98, Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics* (joint with the 17th International Conference on Computational Linguistics), pages 1150–1156.

John R. Searle. 1975. Indirect Speech Acts. In P. Cole and J.L. Morgan, editors, *Syntax and Semantics 3. Speech Acts*. Academic Press. Reprinted in *Pragmatics. A Reader*, Steven Davis editor, Oxford University Press, 1991.

Riccardo Serafin. 2003. Feature Latent Semantic Analysis for dialogue act interpretation. Master's thesis, University of Illinois - Chicago.

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

Douglas M. Towne. 1997. Approximate reasoning techniques for intelligent diagnostic instruction. *International Journal of Artificial Intelligence in Education*.

Peter Wiemer-Hastings. 2001. Rules for syntax, vectors for semantics. In *CogSci01, Proceedings of the Twenty-Third Annual Meeting of the Cognitive Science Society*, Edinburgh, Scotland.