

Dialogue Act Classification, Higher Order Dialogue Structure, and Instance-Based Learning

Barbara Di Eugenio

*Department of Computer Science
University of Illinois at Chicago
Chicago, IL, 60607, USA*

BDIEUGEN@UIC.EDU

Zhuli Xie

*Yahoo! Marketplace Operations
Burbank, CA, 91504, USA*

JACKXIE006@GMAIL.COM

Riccardo Serafin

*TSB
Soluciones Tecnológicas para la Salud y el Bienestar S.A.
Paterna, Valencia, 46980, SPAIN*

BARAKKA@GMAIL.COM

Editor: Massimo Poesio

Abstract

The main goal of this paper is to explore the predictive power of dialogue context on Dialogue Act classification, both as concerns the linear context provided by previous dialogue acts, and the hierarchical context specified by conversational games. As our learning approach, we extend Latent Semantic Analysis (LSA) as Feature LSA (FLSA), and combine FLSA with the k-Nearest Neighbor algorithm. FLSA adds richer linguistic features to LSA, which only uses words. The k-Nearest Neighbor algorithm obtains its best results when applied to the reduced semantic spaces generated by FLSA. Empirically, our results are better than previously published results on two different corpora, MapTask and CallHome Spanish. Linguistically, we confirm and extend previous observations that the hierarchical dialogue structure encoded via the notion of game is of primary importance for dialogue act recognition.

Keywords: Dialogue Acts, Latent Semantic Analysis, k-Nearest Neighbor, Dialogue Games

1. Introduction

Dialogue Act classification is an important task that dialogue systems need to perform in order to understand the role the user's utterance plays in the dialogue, and to generate an appropriate next turn. Dialogue Act (DA) classification can be seen as an operationalization of the venerable debate in linguistics regarding direct and indirect speech acts (Austin, 1962; Searle, 1965, 1975). DA classification was a hot topic of research for about ten years, from the beginning of the 1990s. Influential publications appeared in those years, ranging from human annotation and possible standards thereof (Carletta et al., 1997; Walker and Moore, 1997; Allen and Core, 1997; Jurafsky et al., 1997), to a variety of empirical methods that exploited those annotations to perform practical DA classification for dialogue systems (Reithinger and Maier, 1995; Samuel et al., 1998; Stolcke et al., 2000; Singh et al., 2002). More recent research on DA classification has continued to experiment with empirical methods, e.g. Support Vector Machines (Fernandez and Picard, 2002; Surendran and Levow, 2006)

or our own preliminary set of experiments with Latent Semantic Analysis (Serafin and Di Eugenio, 2004). More interestingly from a linguistic point of view, researchers have been trying to identify which types of features are more predictive, and specifically, whether the dialogue context affects DA recognition (Poesio and Mikheev, 1998; Hastie et al., 2002; Webb et al., 2005; Hoque et al., 2007; Bangalore et al., 2008).

In this paper, we continue in the empirical tradition by exploring Latent Semantic Analysis (LSA) for DA classification on two different corpora, MapTask and CallHome Spanish. We propose Feature LSA (or FLSA) as an extension to LSA, and we show that the reduced spaces built via FLSA improves classification performed via k-Nearest Neighbor (k-NN). The results we obtain are among the best reported in the literature. More importantly, these results offer insights into which features affect DA classification. Our strongest result, that holds across the two corpora, is that classification is aided by the higher structure of the dialogue, encoded via the notion of *conversational game* (Carlson, 1985). Games consist of exchanges whose internal structure is known to and exploited by the participants: paradigmatic examples are question-answer and initiation-response (conversational games are only indirectly related to the notion of games from game theory (Owen, 1995)). Additionally, our best results make no usage of syntactic features such as the syntactic type of an utterance. The claim that the higher order structure of the conversation is an important component of human competence and performance goes back to influential papers such as (Sinclair and Coulthard, 1975; Carlson, 1985; Grosz and Sidner, 1986). However, the work that provides empirical evidence for the importance of higher order dialogue information for DA classification is mostly limited to MapTask (Poesio and Mikheev, 1998; Hastie et al., 2002). We replicate and improve on their results. Importantly, we show that the same holds on a very different corpus in a different language, CallHome Spanish.

As concerns the specific methods we use, first, we propose FLSA, namely, we propose that an LSA semantic space can be built on the basis of arbitrary features, including features that refer to context. Our results confirm that FLSA soundly beats pure LSA for DA classification, since no matter which features we use, results with FLSA are always better than with LSA. Using heterogeneous features is a common practice in other fields, such as image processing, that use Principal Component Analysis or Singular Value Decomposition (the algebraic techniques at the core of LSA): an image can be described by heterogeneous features such as color, texture, shape, and even text annotations. However, in Computational Linguistics, little previous work on LSA has used features other than words. For example, (Wiemer-Hastings, 2001; Kanejiya et al., 2003) add part of speech (POS) tags and some syntactic information to LSA, with mixed results; (Ando and Zhang, 2005) add POS tags, and bigrams that combine both words and their labels, to their theoretical framework that uses SVD.

Second, in the course of our work, we realized that, from a machine learning point of view, LSA can be considered as an *instance-based* learning method. The algebraic transformation that LSA performs is independent from any classification, and it is only when a new data point needs to be dealt with that classification of the data in the training set is used. The simplest instance-based learning method is k-NN. From a practical point of view, we show that the real strength comes from marrying FLSA and k-NN. Our hypothesis, later confirmed by our results, was that the reduced spaces (F)LSA generates would first, improve classification, and orthogonally, allow to train the classifier at much higher speeds.

We present one set of results on the MapTask corpus, and two sets on the CallHome corpus. In both cases, our results are better than any published results we know of. As concerns FLSA

improving the quality of the data space, results on CallHome confirm that k-NN performs best on the reduced space. This holds on MapTask as well, even if, for the best models, differences in performance between k-NN operating on the original space or on the reduced one are statistically indistinguishable. Using FLSA as a preprocessing step to k-NN reduces training time by one or two orders of magnitude – even with current computational power, this is a significant difference.

One important disclaimer before proceeding. We are not claiming that FLSA+k-NN is the best method for DA classification. Doing so would require a direct comparison between FLSA+k-NN and many other learning algorithms, which is beyond the scope of this work. Also beyond the scope of this paper is to evaluate whether the reduced spaces built by FLSA would be effective as input to learning algorithms different from k-NN. We believe our results are strong enough, both per se and as compared to what exists in the literature, to suggest that this is a viable, effective method. We would be thrilled if other researchers were to further explore whether FLSA can improve the performance of other learning algorithms.

The paper is organized as follows. In Section 2, we present the two corpora we use. In Section 3, we introduce the learning methods we adopted, first LSA and FLSA, and then k-NN. In Section 4 we present our experimental setup, and in Section 5, our results. Section 6 is devoted to discussion and comparison with related work, and Section 7, to conclusions and future work.

2. Corpora

We report experiments on two corpora, MapTask and Spanish CallHome. We chose these two corpora because of the different genres they represent, and because they are both annotated for different phenomena, including multiple levels of dialogue structure, from dialogue acts to games to transactions. Additionally, from a practical point of view, they are readily available, and at least for MapTask, there are a number of previous results to compare one’s work with. We will now describe the corpora and the manually annotated features they include. In all the experiments we will describe, we use only gold-standard annotated features.

2.1 The HCRC MapTask corpus

This corpus is a collection of dialogues regarding a “Map Task” experiment. Two participants sit one opposite the other and each of them receives a map, but the two maps differ in which landmarks are included in each. Further, the *instruction giver*’s map has a route indicated while the *instruction follower*’s map does not include the drawing of the route. The task is for the instruction giver to give directions to the instruction follower, so that, at the end, the follower is able to reproduce the giver’s route on her map. The participants know the maps differ, but they don’t know how.

The MapTask corpus is composed of 128 dialogues, for a total of 1,835 unique words and 27,084 dialogue acts. It has been tagged at various levels, from POS to disfluencies, from syntax to DAs (Carletta et al., 1996, 1997).

Utterance Level Features. Among the many features each utterance is annotated for, from prosody to syntax, the ones we use in this work are: *Duration* (a continuous value we discretize into sixteen intervals, as we discuss in Section 4.1); *Who*, a binary valued feature (*Giver/Follower*), that identifies the speaker of the utterance; POS tags for each word; and *SRule*, which indicates the main structure of the sentence, with values *Declarative*, *Imperative*, *Inverted*, and *Wh-Question*.

| Move | Giver | | Follower | |
|----------------------------|-----------------|---|-----------------|---|
| Game 1 – <i>Instruct</i> | | | | |
| 1.a | <i>Ready</i> | right | | |
| 1.b | <i>Instruct</i> | you go up you go south i mean you go north ... up past it ... on the on its right its left-hand side | | |
| Game 1.1 – <i>Query-yn</i> | | | | |
| 1.1.a | | | <i>Query-yn</i> | so i'm just going to be going past the site of the forest fire? |
| 1.1.b | <i>Reply-w</i> | it's just about just below it just below | | |
| 1.1.c | <i>Instruct</i> | it on my map | | |
| Game 1.2 – <i>Check</i> | | | | |
| 1.2.a | | | <i>Check</i> | until just below? |
| 1.c | <i>Instruct</i> | so you just go past the adventure playground on the its left-hand side | | |
| Game 1.3 – <i>Check</i> | | | | |
| 1.3.a | | | <i>Check</i> | and no more aye? |
| 1.3.b | <i>Reply-y</i> | and no more | | |

Table 1: Fragment of a MapTask dialogue

Dialogue Act Coding. The MapTask coding scheme uses 13 DAs (called moves), that include 6 initiating moves, 5 response moves, the *ready* move, and the *other* move.¹ Some of these moves are illustrated in the dialogue fragment in Table 1, taken from <http://www.hcrc.ed.ac.uk/maptask/interface/demo.html>.

The 6 initiating moves are: *Instruct* (a request that the partner carry out an action); *Explain* (one of the participants states some information that was not explicitly elicited by the partner); *Check* (a request to confirm some information that was previously stated); *Align* (an align move checks the attention or agreement of the partner, or her readiness for the next move); *Query-yn*; *Query-w*. The 5 response moves are: *Acknowledge*; *Reply-y*, *Reply-n*, *Reply-w*; *Clarify*, an answer to a question in which the speaker tells the partner something over and above what was strictly asked. The *Ready* move is called a pre-initiating move, it occurs after the close of a game, and serves the purpose of preparing the conversation for a new game to be initiated. Very often *Ready* moves consist of utterances such as “ok” and “right”.

Higher Order Dialogue Coding. The notion of game embodies the hypothesis that participants in a dialogue tend to act according to conventional routines, that generate expectations about what will happen next (Carlson, 1985). In MapTask, a game is defined as *a sequence of moves 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, p. 22). Each of the 6 initiating moves can potentially start a game by providing a purpose to the

1. The MapTask coding manual (Carletta et al., 1996) discusses 12 moves, without mentioning the *other* move. However, 1.2% of the utterances are labelled *other*, hence, the total number of moves in MapTask is 13.

following subdialogue. For instance, instructions signal that the speaker wants the hearer to execute the request, queries signal that the speaker intends to acquire the information requested, and statements signal that the speaker intends the hearer to acquire the given information. A game is tagged with three features: the game purpose, which is the same as its initiating move; where the game ends, or is abandoned; and whether the game either occurs at the top level or is embedded (at some unspecified depth). Note that, even if each game has an initiating move, not all initiating moves begin new games, sometimes they are used to re-state the main goal of the game, etc.

Table 1 shows game 1, the main game, subdivided into several sub-games, games 1.1, 1.2, 1.3 and 1.4. A sub-game may end, the main game may resume and then a new sub-game may begin (i.e., move 1.c that belongs to game 1). Note that Game 1 is an *Instruct* game, not a *Ready* game even if its first move is *Ready*, because the game label is taken from its first *initiating* move, and *Ready* is not an initiating move.

Finally, MapTask dialogues are coded at the transaction level, which provides the subdialogue structure of a complete task-oriented dialogue. Each transaction is built up of several dialogue games and corresponds to one step of the task. We did not use the transaction level in our experiments.

2.2 The Spanish CallHome corpus

This corpus comprises 120 unrestricted phone calls in Spanish between family members and friends, for a total of 12066 unique words and 44628 dialogue acts (Levin et al., 1998; Ries, 1999). Like MapTask, the Spanish CallHome corpus is annotated at a variety of levels, at least for dialogue information.

Utterance Level Features. There are no specific utterance level features, such as prosodic features, POS or syntactic structures, marked on the version of CallHome Spanish we had access to (the CallHome Spanish Dialogue Act Annotation Corpus from the LDC). One available feature is which speaker uttered the utterance, labelled as *Channels* – note that in each conversation there may be more than two speakers, since at the home end more than one person may come to the phone. We defined a few other utterance level features that we used in our experiments, specifically: *DialogueStart*, which marks the first DA in a dialogue; *StopWords* encodes whether the utterance contained any stopwords that were removed (we use a small list of stop words, since our experiments show that even those help recognition);² *Ya*, which indicates that in fact the word *Ya* was taken off the list of stop words and used in the matrix (we observed *Ya* was a good indicator of *back-channels*).

Dialogue Act Coding. The DA annotation is based on the SWBD-DAMSL annotation (Jurafsky et al., 1997). It defines 8 basic categories, *Questions*, *Answers*, *Agreement/Disagreement*, *Discourse Markers* (including *Back-Channels*), *Forward Functions*, *Control Acts*, *Statements* and *Other*. Then, basic tags such as *Statement* can be augmented along several dimensions, such as whether the statement describes a psychological state of the speaker. This results in 232 different dialogue act tags, many with very low frequencies. In this sort of situation, 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 back-channels, obtaining 37 different tags. CallHome37 maintains some subcategorizations, e.g. whether a question is yes/no or rhetorical. In CallHome10,

2. The feature *StopWords* was not defined for MapTask.

| Game | Initiative | Dialogue Act | Speaker | Sentence |
|------------------------------------|------------|-----------------|---------|---|
| Seeking Information (abandoned) | I | wh question | B | pero como, <i>but how</i> |
| Seeking Information | I | yes-no question | B | pero pagan impuestos, <i>but are they taxed</i> |
| | I | statement | B | pero se supone que el menaje no paga <i>but household items are not supposed to be taxed</i> |
| | R | yes answer | A | si' <i>yes</i> |
| Giving Information | I | statement | A | no si' paga impuestos, <i>no yes it is taxed</i> |
| | I | statement | A | paga el quince por ciento, si' señõr <i>it's taxed fifteen per cent, yes sir</i> |
| | R | back-channel | B | ah si' <i>oh yes</i> |

Table 2: Fragment of a CallHome Spanish dialogue

we further collapsed these categories. CallHome10 is reduced to the basic 8 dialogue acts listed above, plus the two tags ' ' for abandoned sentences and ' ' for noise.

Higher Order Dialogue Coding. CallHome Spanish is further annotated for dialogue games and activities. The dialogue game annotation is based on the MapTask notion of a dialogue game, as discussed above. Whereas in MapTask the game label is the same as that of its initiating move, CallHome Spanish uses eight different game labels, *seeking information*, *giving information*, *giving directive*, *action commit*, *giving opinion*, *expressive*, *seeking confirmation* and *communication filler*. The game boundaries are defined by two means: a change of the speaker who has the initiative or a change in the intention of the speaker holding the initiative. Each game consists of a required Initiative Move by one speaker, a Response Move by the other speaker (required or optional depending on the type of game), a Feedback Move by the first speaker (always optional) and a possible second Feedback Move by the second speaker (also always optional). Moves correspond to a single or more DAs, and each is tagged as Initiative, Response or Feedback. All the moves within the same game are labelled with the same game label, from where the game initial move was detected up to a point where either the game was fulfilled or another game was initiated.

The highest level of annotation defines *activities*, which encode the main goal of a certain discourse stretch, such as *gossip* or *argue*. As with MapTask, we did not use this type of information in our experiments.

An example of a dialogue fragment tagged with CallHome Spanish tags is shown in Table 2. The example is taken from (Levin et al., 1998), however the game and dialogue act labels have been expanded from the codes used in that manual for the sake of exposition. Moreover, the dialogue act labels correspond to our groupings of dialogue acts in CallHome37. The Initiative values, *I* and *R*, represent Initiative and Response.

3. Instance Based Learning Methods

On the empirical side, our work on dialogue act classification was motivated by the desire to explore LSA and possible extensions thereof, and evolved into an investigation of LSA as an instance-based learning model, and of other models of this type.

(Mitchell, 1997, Ch. 8) defines instance-based learning methods as those that store all the training data they have; learning takes place when a new data point is presented to the learner. At that stage, a set of similar examples is used to classify the new data point. Clearly, the intelligence of the learner resides in the way that the set of similar examples is computed. With respect to supervised methods, one advantage of instance-based learning methods is that they can construct a different approximation to the function to be learned for each data point in the training set.

Probably the simplest instance-based learning algorithm is *k-Nearest Neighbors* (k-NN). The parameter k determines the cardinality of the set of similar instances. If k is higher than 1, the new query is classified according to the label that appears most frequently among those k training instances. Similarity can be measured in a variety of ways, starting with the basic Euclidean distance. In this paper, we will use cosine similarity, since this is the similarity measure used in LSA.

LSA did not start as a Machine Learning method per se, since in essence LSA applies an algebraic transformation to a vectorial space. However, LSA has often been used for learning, in that the semantic spaces it builds have been used to perform classification of datapoints in a test set, e.g. by assigning to the test datapoint the classification of the closest training point in the semantic space. For this reason, LSA has been called an unsupervised learning method. It appears to us characterizing LSA as an instance-based learning method is more accurate, since class labels of the training data are actively used in classification, precisely as done in k-NN.

The main disadvantage of instance-based learning methods is that classification can take a long time, as opposed to supervised methods in which the model is possibly learned with high cost, but then classification is fast. This may be a problem if an instance-based learning method needs to be used in real time, for example in an actual dialogue system. We will show that applying the space reduction used by LSA to the training data before applying k-NN results in classification which is one or two orders of magnitude faster.

3.1 LSA

Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997; Landauer et al., 1998) is an effective technique to deal with sparse linguistic data spaces, since it prunes the space to preserve only the strongest associations between features. In NLP applications, these features are most often words; the pruning mechanism is based on the algebraic technique of Singular Value Decomposition (SVD), a generalization of Principal Component Analysis which is broadly used in many areas of signal processing (Deprettere, 1988; Kalman, 1996).

In the standard version of LSA, the input is a Word–Document matrix W with a row for each word, and a column for each document. In the majority of applications, documents are extended texts, such as chapters from books or, on a smaller scale, paragraphs. For us, a document is a unit such as an utterance tagged with a dialogue act: we use *utterance* as a generic term for such a unit, which in some cases would more accurately be called a subutterance. In FLSA, we add extra features to LSA by adding a new “word” (row) for each value that the feature of interest can take. For example, in the MapTask corpus each utterance is uniquely associated with either the information giver or the information follower (the tag *Who*). As a consequence, each utterance can

be thought of as having an additional word that specifies who the speaker is. Table 3 shows a portion of the modified Word–Document matrix including part of the dialogue in Table 1.³

Cell $c(i, j)$ contains the frequency with which $word_i$ (either lexical token or feature value) appears in $document_j$. In most work that uses LSA, word frequencies in the matrix are weighted according to specific functions – for example, *tf/idf*, or, as in (Landauer et al., 1998), by entropy. Perhaps surprisingly to the reader, we use raw frequencies. First of all, as much as weighting of word frequencies has become the norm in LSA-based applications, it was initially advocated without much direct experimental support, but by analogy to information retrieval (Landauer et al., 1998, p. 17):

Transforms of this or similar kinds [entropy] have long been known to provide marked improvement in information retrieval, and have been found important in several applications of LSA. They are probably most important for correctly representing a passage as a combination of the words it contains because they emphasize specific meaning-bearing words.

Second, and more importantly, we did run an initial set of inconclusive experiments on the MapTask corpus. In each of these 11 experiments, we ran LSA on the MapTask Word–Document matrix where frequencies were weighted according to one of the weighting schemes implemented in the SMART information retrieval system (Salton, 1971). Such schemes systematically combine term-weighting, document-weighting, and normalization of the subvectors.⁴ We then compared these results to the result we obtained without weighting (specifically, the 43.98% accuracy reported in Table 7, row *MapTask*, column *LSA*). Using raw frequencies was always better than each of 6 different weighting schemes, no matter the dimension r of the reduced matrix (see below); for the other 5 weighting schemes, results were mixed, depending on the specific dimension of the reduced matrix. The bottom line is that none of the weighting schemes we used was clearly better than using raw frequencies, in fact, no weighting scheme looked promising and worth of further inquiry. Hence, we opted for not using weighting schemes and hence for using raw frequencies (we did remove stop words, as we will discuss later).

Whatever transformation is (or is not) applied to the Word–Document matrix, next, LSA applies SVD to it. Any rectangular matrix, for example the $(w \times d)$ matrix of words and documents W , can be decomposed into the product of three other matrices:

$$W = T_0 S_0 D_0^T \tag{1}$$

such that T_0 and D_0 have orthonormal columns and S_0 is diagonal. This is called the *singular value decomposition* of W .

The power of SVD is that it is the basis for an optimal approximate fit using smaller matrices. If the singular values in S_0 are ordered by size, the first r largest are kept and the remaining smaller ones set to zero. The product of the resulting matrices is a matrix \hat{W} of rank r which is only approximately equal to W . It can be shown that the new matrix \hat{W} is the matrix of rank r with the best possible least-squares-fit to W . Since zeros were introduced in S_0 , the representation can be simplified by deleting the zero rows and columns of S_0 to obtain a new diagonal matrix S , and then

3. In the dialogue in Table 1, there are three occurrences of *go* and two of *going*. Since preliminary experiments showed that stemming did not help, each word form is a separate entry in our W matrices.

4. Details can be found at <http://people.csail.mit.edu/jrennie/ecoc-svm/smart.html>.

deleting the corresponding columns of T_0 and D_0 , resulting in:

$$\hat{W} = TSD^T \quad (2)$$

The number of dimensions r retained by LSA is an empirical question. However, crucially r is much smaller than the dimension of the original space.

Once the r -dimensional semantic space has been obtained, when a new document needs to be classified, its vector representation in this space is computed by multiplying its term-vector representation by T and by S^{-1} , the inverse of S . This representation is then used to compare it to the vector of each document in the training set. The tag of the document which has the highest similarity with the 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 an utterance to be tagged with a dialogue act label, and we assign to it the dialogue act label of the document in the training set to which the new document is most similar.

4. Methods

We ran a variety of experiments for the three DA classification tasks of interest, one on MapTask, and two on CallHome Spanish (CallHome37/10). All our experiments use 5-way cross-validation, and all our experiments use the gold-standard features that had been manually annotated on the corpora. In short,

1. We built semantic spaces with LSA and FLSA (each different set of features used for FLSA results in a different semantic space).
2. We used those semantic spaces for classification: for each dialogue act in the test set, we compute its representation in the semantic space, then compute the cosine similarity between this representation and all the other dialogue acts. The new DA will be given the label of the closest DA in the semantic space. These classification experiments also allowed us to determine the most effective number of retained dimensions r , for each feature space.
3. We repeated the experiments with k-NN, for each corpus, and for each set of features attempted with FLSA, under two settings. In setting A, k-NN operated directly on the original training set; in setting B, k-NN operated on the reduced semantic space obtained with (F)LSA that yielded the best accuracy for that corpus and set of features.

We will now provide more details on all three steps in our methodology.

4.1 FLSA in detail

Earlier we noted that the idea behind FLSA is simple: we add extra features to LSA by adding a new “word” for each value that the feature of interest can take. 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, an appropriate count different from zero (most often, simply one) is placed in the new rows if that particular feature value applies to the document under analysis.

For example, in the MapTask corpus each utterance is uniquely associated with either the information giver or the information follower (the tag *Who*). As a consequence, each utterance can be thought of as having an additional word that specifies who the speaker is. Table 3 shows a portion of the modified Word–Document matrix including part of the dialogue in Table 1.⁵ For CallHome Word–Document matrices, since e.g. *Initiative* can take 3 values, we add three new rows, one for each of the three values, Initiative proper, Response and Feedback.

| | 1.a | 1.b | 1.1.a | 1.1.b | 1.1.c | 1.2.a | 1.c |
|------------|-----|-----|-------|-------|-------|-------|-----|
| 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 |
| <Giver> | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| <Follower> | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

Table 3: A portion of the MapTask Word–Document matrix W augmented with speaker identity

What we have described so far rests on two assumptions: that the feature in question only has discrete values, and that the values are mutually exclusive. This is not always the case. For example, in MapTask we used the continuous feature *Duration*: using the exact value obviously results in adding a new row for almost each DAs (we experimented with this approach and obtained poor results, as expected). Rather, we discretized *Duration* into 16 ranges, constructed as follows. We observed that the majority of DAs in MapTask has a duration less than 1 second, the second largest group has a duration between 1 and 2 seconds, and the smallest third group is composed of DAs longer than 2 seconds. Therefore, in order to use a number of categories that was neither too big (leaving the space too sparse) nor too small (not distinguishing useful differences) we used a classification with finer sampling size of 10 subranges between 0 and 1 second, a broader sampling size of 5 subranges between 1 and 2 seconds, and only one category for DAs longer than 2 seconds. This is the classification that was used for the results we are going to present. A couple of different discretization schemes proved not to be useful.

Adding POS tags results in features that do not have mutually exclusive values. We do not mean adding a row for each POS tag and marking its frequency in a sentence, but rather, using POS tags to disambiguate words (Wiemer-Hastings, 2001). This means that it is necessary to add a different row for each distinct pair *word-POS* – for example, in MapTask *route* becomes split as *route-NN* and *route-VB*. Thus, adding POS tags does not result in adding only k rows, where k is the number of possible POS tags; rather, it sizably increases the size of the Word–Document matrix, e.g. for MapTask, from 1,835 for LSA to 2,324 for FLSA.

One last observation on how to add the previous DA tag to the Word–Document matrix. Recall that for us each document (column) is an utterance labelled with a DA. Adding the previous DA

5. In the dialogue in Table 1, there are three occurrences of *go* and two of *going*. Since preliminary experiments showed that stemming did not help, each word form is a separate entry in our W matrices.

is then straightforward. We add d new rows, where d is the number of DAs in the coding scheme (13 for MapTask, 37 or 10 for CallHome, depending on the experiment): a cell in one of these rows will be set to 1 if DA da labels the utterance u_{i-1} preceding utterance u_i , i.e., the column. For example, in Table 1 utterance 1.1.a is preceded by 1.b, labelled as *Instruct*: hence, the cell labelled [Instruct,1.1.a] will be set to 1, and the remaining 12 cells [$da,1.1.a$] will be set to zero, where $da \neq \textit{Instruct}$. Additionally, we experimented with longer histories, with $2 \leq n \leq 4$. To embody a history longer than 1, the Word–Document matrix is built replicating each tag n times: the rows entries embody a concept akin to *tag-n*, i.e., such tag appeared n moves before the current document. As we will see, and had already been observed in the literature, histories of length greater than one did not help.

4.1.1 HOW TO SELECT FEATURES FOR FLSA

An important issue is how to select the features to be added to the Word–Document matrix. One possible answer is to exhaustively train every model that corresponds to one possible feature combination. The problem is that using instance-based learning is in general time consuming for classification. For example, most of our experiments on CallHome10 were run on Intel Xeon workstations with 4 2.4GHz processors, and with 4GB RAM: one full experiment on one set of features with k-NN (5-way cross-validation, 4 possible values for k), took as long as 25 days! Whereas experiments could have been run faster if we had access to supercomputers, and even if contemporary desktops are more powerful than those we used in the last couple of years, combinatorics cannot be escaped in the long run. Hence, we decided to focus only on the most promising models. We evaluated how effective each feature or feature combination may be via *Information Gain* (IG), from decision tree learning (Quinlan, 1993). The IG of a certain feature or feature combination \mathcal{F} measures the reduction in entropy (uncertainty) in the corpus resulting from knowing \mathcal{F} . For example, if for each utterance we know *Who* spoke it, the total entropy of MapTask with respect to DA classification (originally 3.38, see Table 4), is reduced by 0.32 (see Table 5). The main issue was how to select feature combinations. First, we computed the entropy of the corpora with respect to the classification induced by the DA tags (see Table 4). We then computed the IG for all individual features (see Tables 5 and 6, both ordered by increasing IG). We then proceeded as follows.

For MapTask, which is smaller, we ran all experiments by adding each individual feature one at a time. To build combinations of features, we eliminated features with $IG < 0.6$, other than *Who* (Speaker), which we felt would be indicative of specific DAs given the nature of the MapTask corpus – e.g. (Hastie et al., 2002) shows that the two distributions of moves for *Giver* and *Follower* are different. For CallHome, which is larger, Table 6 shows two gaps in the IG values: one between *StopWords* and *PreDA* (larger on CallHome37), and one between *PreDA* and *Game*. We ran experiments adding individual features from *PreDA* on; we included *PreDA* because much previous discourse research attests to its importance.

| Corpus | Entropy |
|------------|---------|
| MapTask | 3.380 |
| CallHome37 | 3.004 |
| CallHome10 | 2.510 |

Table 4: Entropy for MapTask, CallHome37, and CallHome10

| Features | IG | Best Result (Method) |
|---------------------|------|--------------------------|
| Who | 0.32 | 60.32% (k-NN) |
| SRule | 0.53 | 55.72% (k-NN+SVD) |
| Duration | 0.53 | 73.43% (k-NN+SVD) |
| PreDA (Previous DA) | 0.63 | 66.70% (k-NN) |
| Game | 1.21 | 73.08% (k-NN+SVD) |
| Game + Who | 1.62 | 77.97% (k-NN+SVD) |
| Game + PreDA | 1.84 | 76.61% (k-NN+SVD) |
| Game + Who + PreDA | 2.04 | 78.76% (k-NN+SVD) |
| All | 3.31 | 77.71% (k-NN+SVD) |

Table 5: MapTask: Features, Information Gain, and Accuracy

| Corpus | Features | IG | Best Result (Method) |
|------------|---------------------------|-------|--------------------------|
| CallHome37 | Channels | 0.015 | |
| | Ya | 0.036 | |
| | DialogueStart | 0.24 | |
| | StopWords | 0.32 | |
| | PreDA | 0.47 | 75.70% (k-NN+SVD) |
| | Game | 0.59 | 76.66% (k-NN+SVD) |
| | Initiative | 0.69 | 76.80% (k-NN+SVD) |
| | Game + Initiative | 1.09 | 80.07% (k-NN+SVD) |
| | Game + Initiative + PreDA | 1.32 | 80.34% (k-NN+SVD) |
| | All | 2.84 | 78.39% (k-NN+SVD) |
| CallHome10 | Channels | 0.007 | |
| | Ya | 0.028 | |
| | DialogueStart | 0.22 | |
| | StopWords | 0.30 | |
| | PreDA | 0.36 | 77.89% (k-NN+SVD) |
| | Game | 0.53 | 79.48% (k-NN+SVD) |
| | Initiative | 0.66 | 79.37% (k-NN+SVD) |
| | Game + Initiative | 1.01 | 82.49% (k-NN+SVD) |
| | Game + Initiative + PreDA | 1.16 | 82.88% (k-NN+SVD) |
| | All | 2.37 | 82.07% (k-NN+SVD) |

Table 6: CallHome37/10: Features, Information Gain, and Accuracy

Tables 5 and 6 also report the best result for each feature, or feature combination, and the method we obtained it with. Performance is measured in terms of classification accuracy. Boldface highlights the best overall result for that corpus. We will provide many more details on the results shortly, but we would like to mention significance now. For MapTask, differences in performance of at least 0.58% are significant ($p \leq 0.05$); for CallHome, both CallHome37 and CallHome 10, differences in performance of at least 0.44% are significant ($p \leq 0.05$). In both cases, we use χ^2 to determine the significance of performance differences. Second, those tables show that in the vast majority of cases, the combination k -NN+SVD wins over both FLSA and k -NN itself. Even if for some feature combinations simple k -NN performs well, the execution times that we will discuss below show that using SVD prior to k -NN is very useful, if not necessary.

On the whole, IG and increment in performance appear to correlate, although there are exceptions. For example, using all features on MapTask almost reduces the entropy to zero, but performance is significantly lower than just adding three features, *Game + Who + PreDA*. A likely explanation is that the amount of new information introduced is rather low and it is overcome by having a larger and sparser initial matrix. Moreover, when performance improves it does not necessarily linearly increase with IG (see e.g. *Duration vs. Game* in Table 5). Nevertheless, it appears that, when a feature or feature combination has a high IG, there is also a high performance improvement. Therefore, IG can be considered as an approximate indicator of feature quality, and used to weed out unpromising features, or to rank features so that we can train first those FLSA models that use the most promising feature combinations.

4.2 Dimensionality of the reduced matrix and classification

We noted above that the LSA / FLSA experiments helped us to determine the optimal number of retained dimensions r , for each set of features. Here *optimal* is the dimensionality that gives us the best performance, and it is the only dimension we used for the k-NN experiments (for reasons of efficiency we discuss below, we had to reduce the parameter space for the k-NN experiments). As the LSA proponents note, the number of retained dimensions r (i.e., the rank of the reduced matrix \hat{W}), can only be set empirically, and it is a parameter in the algorithm that needs to be provided as input. We experimented with 9 possible values for r between 25 and 350, in increments of 25 between 25 and 100, and of 50 between 100 and 350. For each set of features we thus found an optimal r (number of retained dimensions), which in our experiments always turned out to be 25, across our three classification tasks. This homogeneity may sound surprising: in preliminary experiments we had found that on few feature spaces, the optimal dimension for MapTask had been slightly higher (50 or 75), but when we repeated those experiments the optimal dimension coalesced to 25, for all feature spaces. The optimal dimension for CallHome was always 25. On DIAG-NLP, a third, much smaller corpus (Serafin and Di Eugenio, 2004) the optimal dimension was higher, $r = 50$. But that corpus has various peculiarities, including the coding scheme we devised, hence it is less usable than MapTask and CallHome; for this reason, we do not discuss it in this paper.

We should note that in general, across the literature on LSA and its applications the reduced semantic space retains between 500 and 1000 dimensions, which is much larger than in our experiments. We believe the larger r is due to two factors. First, those applications use much larger corpora than ours, and in addition those corpora sometimes contain less homogeneous information; second, they aim at more “holistic” classification tasks. For example, when LSA is used for information retrieval, the best matching document is not one single utterance tagged with a dialogue act, but a text that may comprise several paragraphs. Further details on all these issues can be found in (Serafin, 2003).

4.3 The k-NN experiments

As mentioned earlier, for each corpus, and for each set of features attempted with FLSA, we repeated the experiments with k-NN, under two settings. In setting A, k-NN operated directly on the original training set; in setting B, k-NN operated on the best semantic space obtained with (F)LSA – where best means the semantic space with the optimal dimensionality r for that specific set of features. As we just discussed, it turns out that $r = 25$ for all corpora, and all sets of features. We call setting B, k-NN+SVD. Whereas it would have in theory been possible to run an exhaustive set of experiments

in which we tried every possible value for reducing the dimensionality r , efficiency considerations prevented us from doing so (see discussion of efficiency for running k-NN in Section 5.2). Please note, the results with (F)LSA per se correspond to k-NN+SVD where $k = 1$. When $k > 1$, we take the DA label as the label of the majority of the k closest DAs, where closeness is still measured via cosine similarity.

Contrary to what claimed by (Hoque et al., 2007), that *extracted speech or discourse features are often projected onto [a] low dimensional subspace*, we are not aware of space reducing techniques being used much if at all in NLP, and certainly almost never for DA classification, one exception being (Fernandez and Picard, 2002).

5. Results

As a preview of the results to be presented, we found that:

- FLSA always beats LSA: namely, adding additional features to a model built only on words always aids classification;
- on CallHome37/10, k-NN+SVD, i.e. k-NN applied to reduced semantic spaces always performs better than k-NN applied to the original training set; on MapTask, k-NN+SVD is better than k-NN, or statistically indistinguishable from it, other than in the 4 worst models, out of 12 we experimented with;
- classification with k-NN on the reduced semantic spaces takes from one to two orders of magnitude less time than on the original training sets;
- on both corpora, models composed of words plus *Game* and *Previous DA* always beat models composed of words plus other features; *Game* appears to be contributing the most to these composite models.

5.1 The LSA and FLSA results

| Corpus | Majority | LSA | FLSA | k-NN+SVD |
|------------|----------|--------|--------|---------------|
| MapTask | 20.69% | 43.98% | 75.94% | 78.76% |
| CallHome37 | 42.68% | 67.15% | 77.74% | 80.34% |
| CallHome10 | 42.68% | 70.53% | 81.27% | 82.88% |

Table 7: Summary of Results

Table 7 gives a summary of the results we obtained. Whereas we include the performance of the majority function as well, we consider LSA as our real baseline.⁶ In all cases, we can see that FLSA always improves performance with respect to LSA for both corpora from about 10-11% for

6. The results of the majority function for CallHome37 and CallHome10 are the same because *statement* is the most frequent DA in both.

CallHome to a dramatic 32% for MapTask. For both FLSA and k-NN, we include the set of features with which the best performance occurs; interestingly, these sets of features coincide for the two methods. Additionally, the best results with k-NN are always obtained on the reduced semantic space, not on the original Word–Document matrix.

All differences in performance between the various methods are highly significant (see above for an explanation of how significance was assessed). Recall that to train and evaluate each method, we used 5-way cross-validation, and gold-standard features. Also recall that the dimension r of the reduced space is always 25, for all methods and corpora, as we discussed earlier.

Table 8 reports a breakdown of the experimental results obtained with FLSA for the three tasks, ordered by increasing performance. All results with FLSA are statistically better than the LSA baseline.

| Corpus | Accuracy | Features |
|------------|----------|---------------------------|
| MapTask | 44.60% | POS |
| | 44.70% | SRule |
| | 49.40% | Who |
| | 53.40% | PreDA |
| | 66.33% | Game |
| | 72.71% | Game + PreDA |
| | 73.50% | Duration |
| | 73.57% | Game + Who |
| | 74.17% | All |
| | 75.94% | Game + Who + PreDA |
| CallHome37 | 70.59% | PreDA |
| | 71.22% | Game |
| | 71.51% | Initiative |
| | 75.48% | All |
| | 76.05% | Game + Initiative |
| | 77.74% | Game + Initiative + PreDA |
| CallHome10 | 73.14% | PreDA |
| | 74.83% | Initiative |
| | 75.30% | Game |
| | 79.68% | Game + Initiative |
| | 79.83% | All |
| | 81.27% | Game + Initiative + PreDA |

Table 8: FLSA Results

5.2 The k-NN results

Tables 9 and 10 report results obtained with k-NN on the three tasks. Results are ordered by increasing accuracy in the k-NN+SVD column; a horizontal line separates individual features from feature combinations, which also result in higher performance. Boldface is used for the best results. We distinguish between “pure” k-NN, where matching is applied to the original training set (column k-NN in the tables), and k-NN applied to the reduced semantic spaces as computed by (F)LSA (column k-NN+SVD) in the tables). k is the number of closest data points k-NN compares the new instance with: we experimented with 4 values for k , 10, 100, 200, 500. Please note, the line *Words*

only should be directly compared with LSA (see Table 7), whereas the other lines can be compared with the FLSA results in Table 8.

| k-NN | k | k-NN+SVD | k | Features |
|---------------|-----|---------------|-----|--------------------|
| 54.17% | 100 | 52.15% | 10 | POS |
| 54.66% | 100 | 52.46% | 100 | Words only |
| 55.72% | 100 | 52.75% | 100 | SRule |
| 60.32% | 10 | 57.57% | 10 | Who |
| 62.95% | 10 | 66.70% | 200 | PreDA |
| 71.30% | 10 | 73.08% | 10 | Game |
| 54.29% | 100 | 73.43% | 10 | Duration |
| 76.60% | 10 | 76.61% | 10 | Game + PreDA |
| 77.19% | 10 | 77.71% | 10 | All |
| 75.52% | 100 | 77.97% | 10 | Game + Who |
| 78.61% | 10 | 78.76% | 10 | Game + Who + PreDA |

Table 9: k-NN and k-NN+SVD success rates on MapTask

| Corpus | k-NN | k | k-NN+SVD | k | Features |
|------------|--------|-----|---------------|----|---------------------------|
| CallHome37 | 64.59% | 100 | 74.63% | 10 | Words only |
| | 66.03% | 10 | 75.70% | 10 | PreDA |
| | 63.64% | 100 | 76.66% | 10 | Game |
| | 68.37% | 10 | 76.80% | 10 | Initiative |
| | 74.59% | 10 | 78.39% | 10 | All |
| | 69.75% | 10 | 80.07% | 10 | Game + Initiative |
| | 71.28% | 10 | 80.34% | 10 | Game + Initiative + PreDA |
| CallHome10 | 67.23% | 100 | 76.95% | 10 | Words only |
| | 68.48% | 10 | 77.89% | 10 | PreDA |
| | 68.19% | 10 | 79.48% | 10 | Game |
| | 71.06% | 10 | 79.37% | 10 | Initiative |
| | 77.97% | 10 | 82.07% | 10 | All |
| | 73.17% | 10 | 82.49% | 10 | Game + Initiative |
| | 74.35% | 10 | 82.88% | 10 | Game + Initiative + PreDA |

Table 10: k-NN and k-NN+SVD success rates on CallHome37/10

We can notice that for CallHome, no matter what the task, or the feature combination, k-NN+SVD is always significantly better than k-NN (recall that any difference of at least 0.44% is significant here). In turn, k-NN+SVD is always better than the FLSA results, see Table 8. For MapTask, at the four lowest levels of performance actually k-NN is more effective than k-NN+SVD. However at higher levels of performance, i.e. on 8 models out of 12, k-NN+SVD is better than k-NN, or the two are statistically equivalent.

However, the question is not only which method is more effective, but which method is more efficient. As we mentioned earlier, we take CallHome10 as our representative task, since MapTask is smaller as a corpus, and the classification on CallHome37 is harder. Each cross-validation run on the original training set on CallHome10, which includes classifying the data according to 4 values of k , 10, 100, 200 and 500, took us from a minimum of 37 hours to a maximum of 25 days! While it is possible to spend weeks and weeks running these experiments as we did, running k-NN on the reduced space decreased the required time for all experiments by one if not two orders of magnitude,

up to a maximum of 26 hours. Note that these times do not include the time it takes to build the reduced semantic space, since that is negligible, and done once and for all. Additionally, while these times were prohibitive for us for running experiments,⁷ they would not be prohibitive to use the model in a real system since only one DA needs to be recognized at a time.

6. Discussion

The strongest conclusion that we can derive from all our results is that the notion of game as embodying higher order discourse structure appears to be really powerful. When looking at Tables 9 and 10, it is striking that the best results in both corpora are obtained with the combination *Game* + *Previous DA* + *third-feature*, where the third feature changes for the two corpora. MapTask and CallHome are two very different corpora, in two different languages: whereas the notion of game in CallHome builds on the MapTask notion of game, the specifics of the actual annotation are quite different. Still, the contribution of higher order discourse structure comes strongly across in both corpora. For MapTask, the notion of game by itself has high information gain (see Table 5), and vastly improves performance by several points with respect to other individual features, other than *Duration*. While performance improves when adding the *Who* and *PreDA* features to *Game*, the contributions of these two features are not as remarkable. The importance of game on MapTask agrees with published results such as (Poesio and Mikheev, 1998; Hastie et al., 2002) but vastly improves on them (please see below for a more detailed comparison).

As concerns CallHome, *Game* still contributes a lot but the other two component features of the best models –*Initiative*, *PreDA*– have a comparable individual performance to *Game*.

The reader may wonder whether the annotation for *Game* in MapTask may introduce circularity, since a game is identified by its initiating move (see Table 1). A reasonable concern is that, when we classify a new utterance e.g. as an *Instruct*, we use redundant information since if this utterance labeled as *Instruct* starts a game, that game will in fact be labeled as *Instruct* as well. A first observation is that, as we discussed in Section 2, not all initiating moves start a game. A stronger counterargument comes from other experiments we ran, with the specific aim of verifying whether any circularity is introduced, and if yes, to what extent. The MapTask corpus has 27084 total moves, of which 9380, i.e. 34%, start a game (at top level or embedded). We used one of the best models we obtained with FLSA for MapTask, which employs *Game* + *Who*,⁸ to separately check the matching rates for initiating and non initiating moves: they are 78.12% and 71.67% respectively. We also trained and tested the FLSA model that uses *Game* + *Who* on only the moves that don't initiate any game, for a matching rate of 71.66% (not even a 2% drop in performance, see Table 8). Hence, we can conclude that, given the notion of game, it is true that initiating moves are somewhat easier to classify than non initiating moves. However, the notion of game is highly beneficial for the classification of non initiating moves as well. Note that the performances on non initiating moves are virtually identical in the model trained on both initiating and non initiating moves, and in the reduced model trained only on non initiating moves. This is probably due to the fact that, although the reduced model includes less data, it is also less sparse.

7. Some of these experiments were run few years ago, without access to supercomputers. Whereas running times would clearly be lower if we ran these experiments today, the combinatorics nature of the problem does not change.

8. We did not use the best model *Game* + *Who* + *PreDA* in order to isolate the effects of the *Game* feature. Since the previous DA of an initiating move is often the end DA of the previous game, we expect that as such it may facilitate classification of the current DA.

As concerns using the game information in real systems to infer DAs, again, an issue of circularity appears to arise, since these two aspects of dialogue structure constrain each other. However, this observation can be turned to advantage, in a model that tries to infer both at the same time, or in a cascaded model. For example, (Hastie et al., 2002) obtains its best results by inferring both move type and move position within a game at the same time. (Bangalore et al., 2008) infers the current dialogue act by using clause level information and the previous dialogue context, which includes game-like information; then, the newly inferred dialogue act, plus lexical information about the current utterance and the previous dialogue context, is used to infer the current game.

As concerns the other features in the best models, they also include the previous DA (even if in CallHome, the model *Game + PreDA + Initiative* is statistically equivalent to *Game + Initiative*). Of course it is not surprising that the previous DA aids in recognition of the current DA. We note that we ran further experiments in which we increased the dialogue history length up to $n = 4$: we found that the higher n , the worse the performance. This agrees with others' observations, e.g. (Ries, 1999; Bangalore et al., 2008).

Finally, the third feature in the best *Game + PreDA + third-feature* models is *Who* (i.e., Speaker) in MapTask, and *Initiative* in CallHome. It is not surprising that speaker information helps in MapTask, since the *Giver* and *Follower* play two different roles in the dialogues, and the distribution of DAs differs among the two (Hastie et al., 2002). Incidentally, it is not surprising that *Channels*, the equivalent feature in CallHome, does not help, since in that corpus the roles of speakers are not differentiated. As concerns *Initiative* in CallHome, it is yet another dialogue context feature that appears to be important for DA classification.

As concerns other features, much work on MapTask uses prosodic information (see below). We only used *Duration*, that we discretized as we described earlier. *Duration* turns out to be a strong predictor with FLSA / k-NN+SVD, notwithstanding its relatively low IG. Further experiments would be needed to tease out the role of *Duration*. We note that our models that do not use prosodic information are better than any published results on MapTask we know of, including those using prosodic information (see Table 11 below).

As concerns syntactic features, to which again we had access only for MapTask, they do not seem to improve performance. This is not surprising, given that it is notorious that syntactic form is not predictive of dialogue acts. In MapTask, *SRule* indicates the main structure of the utterance, *Declarative*, *Imperative*, *Inverted* (i.e. yes/no question), *Wh-question*. POS tags don't help either,⁹ possibly because of the sparser Word–Document matrix that results from the inclusion of POS-tagged words, as we explained earlier. In fact, it had already been observed by (Wiemer-Hastings, 2001) that POS tags are detrimental to performance for LSA based models. In that paper, the task was to interpret the student's input in an Intelligent Tutoring System (Graesser et al., 2000), by comparing the input with stored questions. Similar negative results have been obtained by (Kanejiya et al., 2003). (Wiemer-Hastings, 2001) also tried a second approach, in which LSA was applied separately to each syntactic component of the sentence (subject, verb, rest of sentence), and a final similarity measure was obtained by averaging out those three measures. The results were better than with LSA. Of course, all 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).

9. The attentive reader will note, we do not report the IG for POS in Table 5. Because of how POS tags are added to the matrix, it is not clear how to compute the IG for this feature.

6.1 Related work

As we just discussed, our best results are obtained with models that include both hierarchical and sequential dialogue structure, respectively via game and the preceding DA. While the power of the preceding dialogue history is well established (and most often, it has been found that a history of one works best), the hierarchical dialogue structure has not been used as much in experimental work, with the exception of research on MapTask. Here we discuss similarities and commonalities with that work; on the empirical side, we will show that we improve on previously published results. In fact, our results in general are the highest published on dialogue act classification on almost any corpora, going back to earlier work such as (Samuel et al., 1998), which obtained 75% accuracy for task-oriented dialogues such as Verbmobil with Transformation-Based Learning; and (Stolcke et al., 2000), which obtained 71% accuracy on transcribed Switchboard dialogues, using a tag set of 42 DAs, and using a combination of HMM, neural networks and decision trees trained on all available features (words, prosody, sequence of DAs and speaker identity). We should mention that DA classification on a portion of the Communicator corpus achieved astounding results, of up to 98.5% accuracy (Prasad and Walker, 2002). This is probably due to this specific evaluation being run on a human-computer, rather than human-human, corpus. In fact, when testing on human-human portions of the Communicator corpus, the accuracy decreased to 55.48%.

Many researchers have worked on recognizing dialogue acts in MapTask, and Table 11 summarizes the results that we discuss here. Earlier research on MapTask achieved a 62.1% accuracy

| | Uses Game | Uses Gold-Standard Features | Accuracy |
|-----------------------------|-----------|-----------------------------|----------|
| (Lager and Zinovjeva, 1999) | | ✓ | 62.1% |
| (Poesio and Mikheev, 1998) | ✓ | ✓ | 57.2% |
| (Hastie et al., 2002) | ✓ | | mid 60% |
| (Bangalore et al., 2008) | ✓ | ✓ | 75% |
| (Hoque et al., 2007) | | ✓ | 70.6% |
| This paper | ✓ | ✓ | 78.8% |

Table 11: Dialogue Act Recognition on MapTask

with Transformation-Based Learning and using single words, bigrams, word position within the utterance, previous DA, speaker and change of speaker (Lager and Zinovjeva, 1999). As far as we know, (Poesio and Mikheev, 1998) was the first attempt to include game information for DA classification in MapTask. (Poesio and Mikheev, 1998) contains only 4 models, the best of which achieves an accuracy of 57.2%, against our best accuracy of 78.76%.¹⁰ It is important to note that their models *only* include dialogue features, and not words like ours do, which may explain the difference in performance. As concerns higher-order dialogue features, two of their 4 models almost directly correspond to one of ours: their *bigram* model of DAs, which corresponds to our model that uses only *PreDA*; and their model which includes *PreDA* plus game information. For them, game information includes the type of game as for us, but also, the position inside the game, which we do not encode. The closest to this second model is our *Game + PreDA*. For this latest model they

10. (Poesio and Mikheev, 1998) report two results, for first hypothesis correct, and first or second hypothesis correct. We are comparing our results with their first hypothesis. While in k-NN one has access to other hypotheses, only the one corresponding to the majority of the *k* closest instances is returned.

obtain 50.63% correctness when returning the first hypothesis, while we obtain 76.61% with our k-NN+SVD model.

A much richer attempt at using game information is described in (Hastie et al., 2002). This paper takes a different approach to the problem, and thus it is not straightforward to compare their results to ours. This work predicts move and position within a game simultaneously, e.g. *Instruct-start* or *Ready-middle* (but not the game type, since this can be inferred if the move has been recognized as starting the game itself). The researchers developed this model, since they found unsatisfactory results when trying to recognize game type and move position inside the game independently from the move itself. (Hastie et al., 2002) reports a variety of results on different versions of their combined model, the best of which are in the mid 60%. We should note that inferred labels are used in the prediction task, namely, they move beyond using gold-standard labels.

(Bangalore et al., 2008) also proposes to use the hierarchical structure of the dialogue to recognize DAs. While the notion of game they use on their own corpus is encoding a task / subtask structure rather than conversational games per se, they also run experiments on MapTask. Their best performance on MapTask is about 75%, and surprisingly, they obtain it when excluding dialogue context, which includes the DA and task / subtask labeling of the previous utterance. This holds for their own corpus as well. We believe one reason may be that they add the contextual information to a rich model of the current utterance, which includes, among others, lexical information in the form of unigrams, bigrams, and trigrams of words, and unigrams, bigrams and trigrams of POS tags. As we discussed earlier, first, POS tags do not seem to improve performance on MapTask, which may hinder the model in (Bangalore et al., 2008) as well. Second, it is possible that by just adding context information to a rich utterance model the space becomes too sparse and performance decreases. In fact, the performance of all our models (FLSA, k-NN, and k-NN+SVD) decreases by about 1-1.5% when we use all available features.

(Hoque et al., 2007) uses discourse features to predict dialogue acts on MapTask. However, their notion of discourse feature does not include game information but only the previous dialogue act, and, curiously for us, POS information (since we do not agree that POS information belongs to discourse). Intriguingly, one feature they refer to is *Probabilistic LSA* values, as the probability of an utterance to belong to a certain dialogue act class. However, they find this feature not to contribute to classification and they don't elaborate on it further. They run a series of experiments with a variety of classifiers; their best results on the same classification task we faced (i.e. 13 dialogue acts), is 70.56%.

As concerns Spanish CallHome, a direct comparison with published work is not as easy as with MapTask. Given the high theoretical number of possible DAs (237), most experimental work collapses them, but often papers leave out important details such as the target classification and features used.

(Ries, 1999) reports 76.2% accuracy by using neural networks augmented with the sequence of the n previous DAs (they also find that a short history of 1 works best). However, (Ries, 1999) does not mention the target classification, and the reported baseline appears compatible with both CallHome37 and CallHome10. All results with our best models (with FLSA, k-NN and k-NN+SVD) are better than Ries'. The training features in (Ries, 1999) include prosody and part-of-speech (POS) tags for words, which we did not have access to for our experiments; however, they do not use game information. Other experiments on CallHome Spanish include (Fernandez and Picard, 2002). They obtain a recognition rate of 47.3% on distinguishing between the 8 fundamental DA types in CallHome Spanish (like our setting CallHome10, but without the tags for abandoned sentences

and noise). While this recognition rate is rather low, it was obtained using prosodic cues alone. (Fernandez and Picard, 2002) was one of the first attempts at using SVMs for DA classification. Interestingly, they propose a method similar to ours, in that they apply SVMs to a reduced space obtained via Principal Component Analysis. As a final note, Chu-Carroll and Carpenter (1999) was one of the first attempts to use SVD for utterance classification, as opposed to classifying whole documents or at least paragraphs within documents. Their classification task concerns routing customer calls in a call center, and is akin to topic extraction more than DA classification.

7. Conclusions and Future Work

In this paper, we have presented experiments on DA classification that improve on results presented in the literature, and are innovative from three different points of view. First, from a linguistic point of view we show that higher order dialogue information, and specifically information about conversational games, helps DA classification. Whereas game information had already been found to be helpful in previous work on MapTask, not only do we improve on those earlier results, but we believe we are the first to show that game information is effective on a corpus other than MapTask. Second, from an empirical point of view, we have presented a novel extension to LSA, that we have called Feature LSA. We are not aware of other research in NLP that shows that FLSA is more effective than LSA. Third, we have shown that in fact, the best results are obtained when FLSA is used as a preprocessing step for k-NN, the simplest among all instance-based methods.

Future work includes further investigating the import of other aspects of higher order dialogue structure: linearly via the prior game, or structurally via embedded games, and moving to a higher level of structure, via *Transactions* in MapTask and *Activities* in CallHome. Additionally, we have started looking at logistic regression as a way to select promising features and feature combinations. In a different line of inquiry, we have started applying FLSA and k-NN+SVD to the problem of judging the coherence of texts. Whereas LSA has already been successfully applied to this task (Foltz et al., 1998), the issue is whether FLSA and k-NN+SVD 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 was partially supported by awards N00014-00-1-0640 and N00014-07-1-0040 from the Office of Naval Research, and by awards IIS 0133123, ALT 0536968 and IIS 0905593 from the National Science Foundation. Most of the work was performed while the second and third authors were students at the University of Illinois at Chicago; Michael Glass (now at Valparaiso University) was instrumental in suggesting FLSA for DA classification. Thanks to HCRC (University of Edinburgh) for sharing their annotated MapTask corpus. Finally, thanks to the three anonymous reviewers and the editor for constructive comments and suggestions.

References

James Allen and Mark Core. Draft of DAMSL: Dialog Act Markup in Several Layers. Coding scheme developed by the participants at two Discourse Tagging Workshops (University of Pennsylvania, March 1996, and Schloß Dagstuhl, February 1997), 1997.

- Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.
- John L. Austin. *How to Do Things With Words*. Oxford University Press, Oxford, 1962.
- Srinivas Bangalore, Giuseppe Di Fabbrizio, and Amanda Stent. Learning the Structure of Task-Driven Human–Human Dialogs. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1249–1259, September 2008.
- Jean Carletta, Amy Isard, Stephen Isard, Jacqueline C. Kowtko, Gwyneth Doherty-Sneddon, and Anne H. Anderson. HCRC dialogue structure coding manual. Technical Report HCRC/TR-82, University of Edinburgh, 1996.
- Jean Carletta, Amy Isard, Stephen Isard, Jacqueline C. Kowtko, Gwyneth Doherty-Sneddon, and Anne H. Anderson. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–31, 1997.
- Lauri Carlson. *Dialogue Games. An Approach to Discourse Analysis*, volume 17 of *Studies in Linguistics and Philosophy*. D. Reidel Publishing Company, 1985.
- Jennifer Chu-Carroll and Bob Carpenter. Vector-based Natural Language Call Routing. *Computational Linguistics*, 25(3):361–388, 1999.
- Ed F. Deprettere, editor. *SVD and signal processing: Algorithms, applications and architectures*. North-Holland Publishing Co., Amsterdam, The Netherlands, 1988. ISBN 0-444-70439-6.
- Raul Fernandez and Rosalind W. Picard. Dialog Act Classification from Prosodic Features Using Support Vector Machines. In *Speech Prosody, ISCA International Conference*, pages 291–294, Aix-en-Provence, France, April 2002.
- Peter W. Foltz, Walter Kintsch, and Thomas K. Landauer. The measurement of textual coherence with Latent Semantic Analysis. *Discourse Processes*, 25:285–308, 1998.
- Arthur C. Graesser, Katja Wiemer-Hastings, Peter Wiemer-Hastings, Roger Kreuz, and the Tutoring Research Group. Autotutor: A simulation of a human tutor. *Journal of Cognitive Systems Research*, 2000.
- Barbara Grosz and Candace Sidner. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12:175–204, 1986.
- Helen Wright Hastie, Massimo Poesio, and Stephen Isard. Automatically predicting dialogue structure using prosodic features. *Speech Communication*, 36(1–2):63–79, 2002.
- Mohammed E. Hoque, Mohammad S. Sorower, Mohammed Yeasin, and Max M. Louwerse. What Speech Tells us about Discourse: The Role of Prosodic and Discourse Features in Dialogue Act Classification. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, Orlando, FL, August 2007.
- Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coders Manual, Draft 13. Technical Report 97-02, University of Colorado, Boulder. Institute of Cognitive Science, 1997.

- Dan Kalman. A Singularly Valuable Decomposition: The SVD of a Matrix. *The College Mathematics Journal*, 27(1):2–23, 1996.
- Dharmendra Kanjejiya, Arun Kumar, and Surendra Prasad. 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, 2003.
- Torbjörn Lager and Natalia Zinovjeva. 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), 1999.
- Thomas K. Landauer and Susan T. Dumais. A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240, 1997.
- Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284, 1998.
- Lori Levin, Ann Thymé-Gobbel, Alon Lavie, Klaus Ries, and Klaus Zechner. A discourse coding scheme for conversational Spanish. In *ICSLP-98, the Fifth International Conference on Spoken Language Processing*, 1998.
- Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.
- Guillermo Owen. *Game Theory*. Academic Press, UK, third edition, 1995.
- Massimo Poesio and Andrei Mikheev. The Predictive Power of Game Structure in Dialogue Act Recognition: Experimental Results Using Maximum Entropy Estimation. In *ICSLP-98, the Fifth International Conference on Spoken Language Processing*, 1998.
- Rashmi Prasad and Marilyn Walker. Training a Dialogue Act Tagger for Human-human and Human-computer Travel dialogues. In *The Third SIGdial Workshop on Discourse and Dialogue*, 2002.
- J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- Norbert Reithinger and Elisabeth Maier. Utilizing statistical dialogue act processing in Verbmobil. In *ACL95, Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, 1995.
- Klaus Ries. HMM and Neural Network Based Speech Act Detection. In *Proceedings of ICASSP 99, The IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, March 1999.
- Gerard A. Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.
- Ken Samuel, Sandra Carberry, and K. Vijay-Shanker. 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, 1998.

- John R. Searle. What Is a Speech Act. In Max Black, editor, *Philosophy in America*, pages 615–628. Cornell University Press, Ithaca, New York, 1965. Reprinted in *Pragmatics. A Reader*, Steven Davis editor, Oxford University Press, 1991.
- John R. Searle. Indirect Speech Acts. In P. Cole and J.L. Morgan, editors, *Syntax and Semantics 3. Speech Acts*. Academic Press, 1975. Reprinted in *Pragmatics. A Reader*, Steven Davis editor, Oxford University Press, 1991.
- Riccardo Serafin. Feature Latent Semantic Analysis for dialogue act interpretation. Master’s thesis, University of Illinois - Chicago, 2003.
- Riccardo Serafin and Barbara Di Eugenio. FLSA: Extending Latent Semantic Analysis with features for dialogue act classification. In *ACL-EACL04, 42nd Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 2004.
- John Sinclair and Malcolm Coulthard. *An Analysis of Discourse: The English Used by Teachers and Pupils*. Oxford University Press, 1975.
- Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. *Journal of Artificial Intelligence Research (JAIR)*, 15:105–133, 2002.
- Andreas Stolcke, Klaus Ries, Noah Coccaro, Elisabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373, 2000.
- Dinoj Surendran and Gina-Anne Levow. Dialog Act Tagging with Support Vector Machines and Hidden Markov Models. In *Proceedings of Interspeech*, pages 1950–1953, 2006.
- Marilyn A. Walker and Johanna D. Moore. Empirical studies in discourse. *Computational Linguistics*, 23(1):1–12, 1997. Special Issue on Empirical Studies in Discourse.
- Nick Webb, Mark Hepple, and Yorick Wilks. Dialogue Act Classification Based on IntraUtterance Features. In *Proceedings of the AAAI Workshop on Spoken Language Understanding*, 2005.
- Peter Wiemer-Hastings. Rules for Syntax, Vectors for Semantics. In *CogSci01, Proceedings of the Twenty-Third Annual Meeting of the Cognitive Science Society*, Edinburgh, Scotland, 2001.