How to Classify Tutorial Dialogue? Comparing Feature Vectors vs. Sequences

JOSÉ P. GONZÁLEZ-BRENES, WEISI DUAN, AND JACK MOSTOW
Language Technologies Institute, Carnegie Mellon University, USA

A key issue in using machine learning to classify tutorial dialogues is how to represent time-varying data. Standard classifiers take as input a feature vector and output its predicted label. It is possible to formulate tutorial dialogue classification problems in this way. However, a feature vector representation requires mapping a dialogue onto a fixed number of features, and does not innately exploit its sequential nature. In contrast, this paper explores a recent method that classifies sequences, using a technique new to the Educational Data Mining community – Hidden Conditional Random Fields [Quattoni et al., 2007]. We illustrate its application to a data set from Project LISTEN's Reading Tutor, and compare it to three baselines using the same data, cross-validation splits, and feature set. Our technique produces state-of-the-art classification accuracy in predicting reading task completion. We consider the contributions of this paper to be (i) introducing HCRFs to the EDM community, (ii) formulating tutorial dialogue classification as a sequence classification problem, and (iii) evaluating and comparing dialogue classification.

Key Words and Phrases: Project LISTEN, Feature Vectors, Sequence Classification, Reading Task Completion

1. INTRODUCTION

Researchers in education have long distinguished a *student trait*, a characteristic that is relatively constant, from a *student state*, a characteristic that changes thorough time [Reigeluth, 1983]. In this paper, we discuss how to train a classifier to represent time-varying characteristics of student states.

We illustrate our discussion with an example. Suppose we are classifying computer-student dialogues using the single feature “turn duration”. Figure 1 shows the duration of each of the turns in a dialogue (9s, 8s, 5s, 7s, and 6s respectively). Conventional classifiers, like logistic regression or decision trees, rely on a fixed-size feature vector as an input; hence, we have to decide *a priori* how many features we are going to include. But, how to map into a fixed-size feature vector a dialogue that may vary in number of turns? One approach is to extract features from a window, either from the beginning or the end of the dialogue [González-Brenes and Mostow, 2011]. There are (at least) two alternative approaches: (i) averaging the value of the features in the window – in our example, it would be a single feature with value 6.0; or (ii) having a feature for every turn – in our example, three features with values 5, 7 and 6. Once we transform dialogues into feature vectors, we can train conventional classifiers on them.

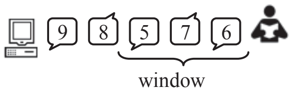


Figure 1: Dialogue described by a single feature

Mapping dialogues into feature vectors does not innately capture or exploit the sequential nature of dialogue. Furthermore, it is not clear how appropriate the window strategy is, since short windows may exclude important information, whereas long windows may have too many missing values. In this paper, we consider the alternative approach of classifying over the entire dialogue using sequences, by applying Hidden Markov Models, and we introduce a recent technique, Hidden Conditional Random Field (HCRF) [Quattoni et al., 2007].

Authors' addresses: J.P. González-Brenes, e-mail: joseg@cs.cmu.edu; W. Duan, e-mail: wduan@cs.cmu.edu; J. Mostow, e-mail mostow@cs.cmu.edu. , Project LISTEN, RI-NSH 4103, 5000 Forbes Avenue, Carnegie Mellon University, Pittsburgh, PA 15213-3890, USA.

The rest of this paper is organized as follows. Section 2 discusses relation to prior work. Section 3 describes the different feature vector and sequence classifiers we consider to classify dialogues. Section 4 presents empirical results on a classification task to predict whether a student will complete a reading task. Section 5 concludes.

2. RELATION TO PRIOR WORK

Previous work on representations of data in language technologies has relied on feature vectors using bag of word representations, *n*-grams, or their projections into latent space [Wallach, 2006]. Alternatively, kernels have allowed richer representations. For example, for text classification, the String Kernel [Lodhi et al., 2002], represents documents in a feature space of all of the substrings of length *k*. A similar feature vector representation would involve a prohibitive amount of computation, since the size of the feature vector space grows exponentially with *k*. Sequence Kernels have been used for speaker verification to map the audio signal sequence into a single feature vector using polynomial expansions [Louradour et al., 2006]. We are unaware of alternative classification approaches for dialogue other than using feature vectors.

Classification of sequences can be categorized in three different ways [Xing et al., 2010]: feature vector based classification, model based classification, and distance based classification. In the rest of this section, we discuss previous approaches to dialogue classification in these categories.

2.1 Feature Vector Based Dialogue Classification

As discussed earlier, sequences can be mapped into fixed-size feature vectors. As far as we know, all of the previous approaches in classification of tutorial dialogue have ignored the sequential nature of dialogue, constraining dialogue into a fixed-size representation. For example, predicting dialogue completion has been studied extensively in the literature, relying on a feature vector representation [González-Brenes et al., 2009; González-Brenes and Mostow, 2010; González-Brenes and Mostow, 2011; Hajdinjak and Mihelic, 2006; Möller et al., 2008; Möller et al., 2007; Walker et al., 2001].

2.2 Model Based Dialogue Classification

Model based classification models sequences directly, for example using Hidden Markov Models (HMMs). In this paper, we advocate for model based approaches over using feature vectors.

HMMs have been used extensively in language technologies, for example in topic segmentation [Eisenstein et al., 2008]. In the dialogue community, to our knowledge, HMMs have been used only to segment dialogue [Stolcke et al., 2000], but not to classify it as we do here. A growing body of work has investigated how to use policy learning to improve tutorial effectiveness [Ai et al., 2007; Beck, 2004; Beck and Woolf, 2000; Boyer et al., 2010; Chi et al., 2008; Chi et al., 2010]. Policy learning often relies on Markov Decision Processes (MPDs) [Singh et al., 1999] to learn a strategy that maximizes the expected value of a specified reward function. MDPs are very similar to HMMs in that the input is a sequence. However, learning a strategy for what to do at each point in a dialogue is a different problem than learning a classifier. Although speech is traditionally modeled as a sequence of phonemes [Gunawardana et al., 2005], we believe we are the first to model dialogues without using feature vectors. We do not know of any previous use of HCRFs in the Educational Data Mining community.

2.3 Distance Based Classification

Distance-based methods for sequence analysis rely on a distance function to measure the similarity between two sequences. Dialogue System Difference Finder [González-Brenes et al., 2009] defines a distance function between dialogues described by feature vectors. We are unaware of distance functions between dialogues that model dialogues as sequences.

3. DIALOGUE CLASSIFICATION

In this section, we discuss the classification algorithms we considered to model tutorial dialogue behavior using either feature vectors or sequences. For feature vector classification we considered Maximum Entropy Classification [Berger et al., 1996] and Random Forest [Breiman, 2004]. We used Maximum Entropy Classification, often called Logistic Regression, as a baseline because of its recent success in classifying tutorial dialogue [González-Brenes and Mostow, 2011]. Random Forest, often called Ensemble of Decision Trees, has provided good empirical results in the EDM community, having being used in the winning submission of the Educational Data Mining Challenge at SIGKDD 2010.

Alternatively, for classifying sequences, we use the popular Hidden Markov Model (HMM) approach [Rabiner, 1989]. We also introduce to the EDM community a recent technique called Hidden Conditional Random Fields (HCRFs), which have been applied to other domains [Gunawardana et al., 2005; Sy Bor, 2006]; for details of their implementation, see [Quattoni et al., 2007].

Maximum Entropy, and HCRF can be formulated under an approach called risk minimization [Obozinski et al., 2007], where the parameters are estimated by maximizing the fit to the training data while penalizing model complexity (number of features). Better fit to the training data favors classification accuracy in the training set, but risks over-fitting the model to the data. Conversely, low model complexity sacrifices classification accuracy on the training set in hopes of generalizing better to unseen data. Both Maximum Entropy and HCRF are log-linear and discriminative – they model the differences between class labels without inferring generative models of the training data. However, they differ in the way they calculate the fit to the training data: HCRFs use a latent variable (a hidden state) to model input sequences, while logistic regression uses feature vectors. To penalize complexity, they both rely on regularization penalties. The two most popular regularization penalties are the L_1 norm and the L_2 norm of the feature vector [Ng, 2004]. The L_1 norm selects fewer features than the L_2 norm, and hence it is used when interpretability of the model is desired, or when the number of features exceeds the number of data points. Conversely, when the number of features is small compared to the training data, the L_2 norm offers better predictive power [Zou and Hastie, 2005]. The trade-off between fit to the training data and model complexity is controlled by a so-called regularization hyper-parameter, often optimized during cross-validation using a held-out set of development data.

Random Forest is an ensemble of decision trees. To avoid over-fitting, each tree is grown using only a random subset of the features and a random subset of the training data. The training procedure grows each tree greedily, selecting the best decision split at each node, and stopping when each leaf has five data points, with no pruning. During testing, Random Forest returns the class predicted by the largest number of decision trees. Random Forest does not assume that the data belongs to any particular distribution, and hence it is considered a non-parametric approach.

An HMM is a generative classifier. Thus to distinguish between two classes, it requires two models: one for the positive class, and one for the negative class. Like an HCRF, an HMM models its input as a sequence, and uses a latent variable to model hidden state. Using hidden variables in HCRFs and HMMs converts the learning problem into non-convex optimization. Consequently, the parameters used to initialize the model in the training procedure affect its final performance.

In Table II, we show a summary of the differences between the classifiers we described. The plate diagrams of Maximum Entropy, HCRFs, and HMMs follow the convention of coloring the circles of the variables that are observable during training. The outcome variable **y** is the class label we want to learn. For example, we may label dialogues with **y=+1** if they were completed successfully, and **y=-1** otherwise. For feature vector classification, **x** is a feature vector describing an entire dialogue. In contrast, for sequence classification, $\mathbf{x_t}$ is the value of the features at time **t**. For example, **x** can represent the duration of each turn in the dialogue. The hidden discrete variable $\mathbf{s_t}$ is not directly observed. Figure 2 expands the plate notation of Table II for HCRFs. The undirected graphical model notation indicates that a variable is independent of all the other variables given its neighbors. For example, the hidden state $\mathbf{s_T}$ is independent of all other variables given **y**, $\mathbf{x_T}$, and $\mathbf{s_{T-1}}$. Instead of drawing each repeated variable, a plate is used to group repeated variables. The class label **y** depends only on the hidden states. The directed graph notation to represent HMMs uses a conventional Bayesian Network representation.

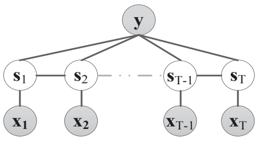
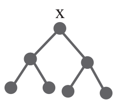
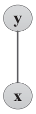
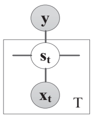
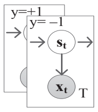


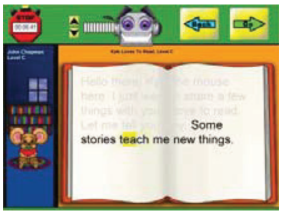
Figure 2: Graphical representation of a Hidden Conditional Random Field

Table II: Bird's-eye View of the Classifiers Considered

	Random Forest 	**Max. Ent.** 	**HCRF** 	**HMM** 
Input	Feature vector	Feature vector	Sequences	Sequences
Classifier type	Non-parametric	Parametric / discriminative	Parametric / discriminative	Parametric / generative
Convex?	No explicit objective function	Yes	No	No
Random?	Subset of training and features	No	Initialization	Initialization
Distribution	Non-linear	Log-linear	Log-Linear (in latent space)	Gaussian
Overfitting protection	Randomization	Regularization penalty	Regularization penalty	Prior on emission distribution

4. EXPERIMENTAL EVALUATION

We compared the methods on data logged by Project LISTEN's Reading Tutor, which listens to a child read aloud, and takes turns picking stories to read [Mostow and Aist, 2001]. The Reading Tutor adapts the Sphinx-II speech recognizer [Huang et al., 1993] to analyze the students' oral reading, and intervenes when it notices the reader makes a mistake, get stuck, click for help, or encounter difficulty. Figure 3 shows a screenshot of the 2005 Reading Tutor. The current sentence is in boldface, and the tutor is giving help on the highlighted word *teach*. The Reading Tutor is an atypical dialogue system, in the sense that it is not designed to answer questions by a user. However, it addresses such dialogue phenomena as turn-taking, backchanneling, mixed initiative, and multimodal interactions (speech and mouse).



Some stories teach me new things.

Figure 3: Project LISTEN's Reading Tutor screenshot

We demonstrate our approach using a previously studied prediction problem [González-Brenes and Mostow, 2010; González-Brenes and Mostow, 2011]. We count the interaction of a student reading a story with the tutor as one dialogue. We consider dialogues to be composed of one or more *sentence encounters* in which the Reading Tutor displays a sentence for the student to read. We want to classify each dialogue at runtime, based on the sentence encounters so far, according to whether the student will finish reading the story or is about to stop reading.

We calculate all features using only information available at prediction time. Thus for positive training examples, we truncate each finished story to a random number of sentence encounters before calculating features. For negative training examples, we use unfinished stories, but we do not truncate them. Table II summarizes the sorts of dialogue features used. We extract features only from the student's sentence encounters, not from the tutor's utterances, because we want to base our predictions on the student's behavior.

Table II: Features considered

Prosodic Features: Various duration, pitch and intensity features, as described in [Duong and Mostow, 2010].
Sentence Features: Properties of a sentence to be read, such as percentage of story read so far, number of word types and tokens, number of clicks for help, and statistics on word length and frequency

4.1 Dataset

The data set we used was logged by Project LISTEN's Reading Tutor while used regularly at elementary schools during the 2005-2006 school year. To obtain a balanced data set of 2,112 dialogues with 162 children, we randomly selected half of them from dialogues where the students completed the reading, and the other half where they did not. We only included dialogues with at least four sentence encounters, to provide some basis for prediction. The selected dialogues average 18 sentences encounters.

Training and testing on the same students can risk relying on peculiarities of individual students. Hence, we separated the data such that the development and testing sets had no students from the training set. We report all results using 10-fold cross-validation across students. Because the folds can vary in size, we report an average weighted by the number of data points in each individual fold. We take this variation into consideration when we report significance in our statistical tests, weighting each fold accordingly, as described previously [Bland and Kerry, 1998]. We split each fold into four non-overlapping sets: training set (70% of the students), two development sets (each with 10% of the students), and test set (with 10% of the students).

4.2 Data preparation

When using HMMs with continuous feature values, the initialization of the parameters is very important. One of the parameters of a continuous HMM, the covariance matrix of the emission probability, cannot be initialized with just any random numbers. In fact, at every point of training, it should adhere to some rules: it must be non-singular (invertible) and positive semi-definite. On preliminary experiments with HMMs, the large size of the feature set relative to the amount of training data resulted in non-singular covariance matrices that assign infinite likelihood to sequences.

To avoid such problems in training HMMs, we reduced the feature space by eliminating features that are "usually the same value." For this purpose we used a statistical property of distributions called kurtosis, also referred as the fourth standardized moment. Gaussian distributions have zero kurtosis, "peakier" distributions than the Gaussian have positive kurtosis, and conversely, flatter distributions have negative kurtosis. To make our results comparable across classifiers, we want all classifiers to have access to exactly the same set of features. Hence, we use the training data in each cross-validation fold to remove features that have kurtosis greater or equal than 100, so all other classifiers are on a level playing field with HMMs. We also perform the standard transformation of centering the feature values as z-scores with mean zero and standard deviation one.

The value of our features changes thorough time. But what's the minimum unit of time? Our methods depend on discrete time-steps, and so we considered the following alternatives: one second, a word uttered by the student, or a sentence encountered. For simplicity we decided to use a sentence encounter as the minimum time unit, since it was the easiest to map from the format of the tutor logs. Hence, to map dialogues into feature vectors, we extract features from a window of *w* sentences from the end of the dialogue. In the case of predicting task completion, the last dialogue turns are the most informative [González-Brenes and Mostow, 2011]. Sequences are computed by calculating the values of the features for each sentence encounter.

4.3 Describing Dialogues as Feature Vectors

In this subsection we explore the shortcomings of using feature vectors to describe tutorial dialogue. For this purpose, we compared the following feature vector classifiers:

- Random Forest, as implemented by the Statistics Toolbox of Matlab. We used 300 decision trees in each forest.
- Maximum Entropy classifier with L_1 and with L_2 regularization. We used PMTK for Matlab [Murphy, 2012, in preparation], February 28th, 2011 release[1].

We now analyze the effect of the window size on classification accuracy. Figure 4 shows the average cross-validation accuracy of the classifiers tested on the Development Set 2. The classifiers were trained using the Training Set portion of each fold. For Maximum Entropy, we tuned the regularization hyper-parameters using the Development Set 1 within each fold independently. The left panel of the figure shows the accuracy of classifiers averaging the values of the features across different window sizes, and the right panel shows the accuracy of classifiers using a different feature at each time step (for example, the "duration" would be represented with different features if it is computed over the last sentence, or the second to last sentence, and so on). All of the classifiers significantly outperform the expected value of a classifier that randomly picks class labels (the "guess" line), as determined by a one-sample *t*-test at the 5% significance level.

[1] http://code.google.com/p/pmtk3/

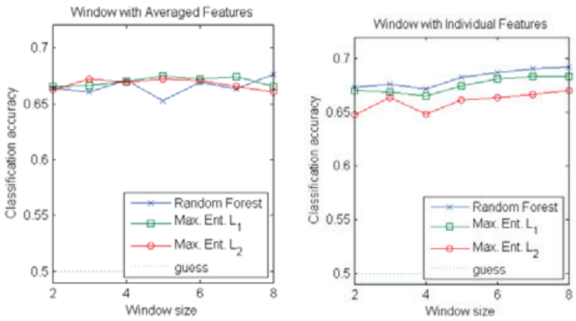


Figure 4: Classification accuracy of different dialogue to feature vector strategies

The strategy of aggregating features achieves approximately 67% classification accuracy across all different window sizes regardless of the classifier used – the differences between classifiers are not significant at $p < .05$. On the other hand, in the right panel of Figure 4, when features are not aggregated within time steps, there is a trend of increasing classification accuracy with larger windows. The best classifiers use the largest window size: Random Forest achieves 69.2% classification accuracy, followed by L_1-regularized Maximum Entropy with 68.3% accuracy and L_2-regularized Maximum Entropy with 67.0% accuracy.

We observe that when modeling individual features for each time step individually, longer windows have better classification accuracy. This finding supports the hypothesis that a representation that includes the whole dialogue is desirable. Because the size of a dialogue is unbounded, it is impossible to define a feature vector that could describe each of the time steps of any dialogue without aggregation. Furthermore, feature vector classifiers do not know that some features represent a value that is changing through time, and hence do not exploit any temporal relation. In the next subsection, we explore a more natural way to model tutorial dialogue as sequences.

4.4 Describing Dialogues as Sequences

We study modeling tutorial dialogue as sequences directly. For this purpose we compared the following sequence classifiers:

- Hidden Markov Models, using the PMTK toolkit mentioned earlier.
- Hidden Conditional Random Fields, using the HCRF Library,[2] version 2.0b. We chose the option of using L-BFGS as the optimizer.

The classifiers were trained using the training set portion of each fold. For HMMs, we used five random restarts, picking the best initialization using Development Set 1. To initialize the parameters, we chose the library's default initialization, which uses a prior to favor a diagonal covariance matrix for the emission probability. We used the conventional Expectation Maximization (EM) algorithm to estimate the transition and emission probabilities. We did not perform random restarts for the HCRFs due to time constraints.

Since sequential models rely on hidden states, we want to understand the effect of the number of hidden states on classification accuracy. Figure 5 shows the average cross-validation accuracy of the classifiers tested on the Development Set 2. For HCRFs, the regularization hyper-parameters tuned were tuned with the Development Set 1 within each fold independently. All of the classifiers significantly outperform the expected value of a classifier that randomly picks class labels (the "guess" line), as determined by a one-sample *t*-test at the 5% significance level. We observe that using L_1 regularization does

[2] http://sourceforge.net/projects/hcrf/

not affect the classification accuracy across the number of hidden states, as it stays relatively constant around 69%. L_2 regularization is more prone to overfitting, and hence the addition of extra hidden states reduces classification accuracy, from 69% with two hidden states, to 65% with three. HMMs are the worst performing models, with a classification accuracy of 61% with two hidden states, which gets a small gain with three hidden states to 62%, and then decreases again to 61% with four hidden states.

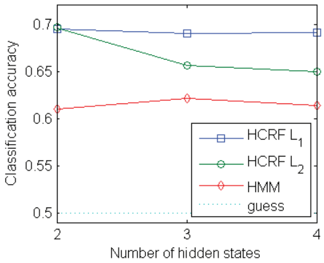


Figure 5: Classification accuracy across different number of hidden states

4.5 Feature Vectors versus Sequences

We now investigate whether classifying over entire sequences offers better classification accuracy than using feature vectors. For this comparison, we select the best classifiers from the two previous subsections, and test them on the unseen test set of each fold. That is, we compare HCRF with L_1 regularization, Random Forest, Maximum Entropy (using individual features for time steps) with L_1 regularization, and HMM.

Table 1 shows the classification accuracy of the best classifiers described earlier with their 95% confidence intervals. We observe that HCRFs using L_1 regularization outperforms all other classifiers, with a classification accuracy of 69.32%. Although the confidence intervals overlap, a *t*-test at the 5% level reveals that HCRFs are significantly better than Maximum Entropy (accuracy = 66.57%) and HMMs (accuracy = 62.50%).

Table 1 Classifier Comparison

	Accuracy	Precision	Recall
HCRF (L1)	**.6932** ± .03	**.6909**	.6890
Random Forest	.6799 ± .03	.6709	.7172
Maximum Entropy (L1)	.6657 ± .03	.6669	.6704
HMM	.6250 ± .03	.5932	.8000
Random Baseline	.5000 ± .03	.5000	**1**

Random Forest is a strong contender, because unlike the other methods we compared, it does not assume any particular distribution of the data. However, a *t*-test reveals that its classification accuracy is not significantly different ($p>0.05$) from the Maximum Entropy baseline used in previous work [González-Brenes and Mostow, 2011]. The *t*-test does not reject the null hypothesis that HCRFs and Random Forest (accuracy = 67.99%) have the same classification accuracy. This finding may suggest that the HCRF model allows more consistent results, but further experimentation is required to understand when and why each method works better than the other one. HCRF and Random Forests took a few hours to train, without making use of the parallelization options available.

5. CONCLUSION

We consider the contributions of this paper to be (i) introducing HCRFs to the EDM community, (ii) formulating tutorial dialogue classification as a sequence classification problem, and (iii) evaluating and comparing dialogue classification algorithms to predict completion of a reading task by children using Project LISTEN's Reading Tutor. A limitation of our approach is that we did not perform explicit feature selection before learning a classifier. This omission may had a negative impact on the classification accuracy of models more prone to over-fitting, particularly HMMs.

Although HMMs can also classify tutorial dialogue using sequences, they do not achieve good classification accuracy, presumably because they do not scale well to large feature sets. HCRF allows state-of-the-art results for predicting reading task completion. Moreover, HCRF allows modeling tutorial dialogues as sequences, which is a more natural representation than feature vectors.

Future work should study how the hidden states of HCRF segment the dialogues. We hypothesize that the hidden states of the model are related to the motivational states of the students. Additionally, we believe that further improvement in classification accuracy could be gained with a model that combines the strength of using a sequence representation with a non-parametric approach such as Random Forest.

ACKNOWLEDGEMENTS

This work was supported in part by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080628 to Carnegie Mellon University. The opinions expressed are those of the authors and do not necessarily represent the views of the Institute or U.S. Department of Education. We thank the educators, students, and LISTENers who helped generate, collect, and analyze our data, and the reviewers for their helpful comments. We appreciate Prof. Louis-Phillipe Morency's help with the HCRF library. José was partially supported by the Costa Rican Ministry of Science and Technology (MICIT).

REFERENCES

AI, H., TETREAULT, J.R. and LITMAN, D.J. 2007. Comparing user simulation models for dialog strategy learning, Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers. Association for Computational Linguistics, Rochester, New York, 1-4.

BECK, J. 2004. Using response times to model student disengagement, Proceedings of the ITS2004 Workshop on Social and Emotional Intelligence in Learning Environments, 13-20.

BECK, J. and WOOLF, B.P. 2000. High-Level Student Modeling with Machine Learning, Proceedings of the 5th International Conference on Intelligent Tutoring Systems. Springer-Verlag, London, UK, 584-593.

BERGER, A.L., PIETRA, V.J.D. and PIETRA, S.A.D. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist. 22*, 39-71.

BLAND, J.M. and KERRY, S.M. 1998. Weighted comparison of means. *British Medical Journal 316*(7125), 129.

BOYER, K., PHILLIPS, R., INGRAM, A., HA, E., WALLIS, M., VOUK, M. and LESTER, J. 2010. Characterizing the Effectiveness of Tutorial Dialogue with Hidden Markov Models, Intelligent Tutoring Systems. Springer Berlin / Heidelberg, 55-64.

BREIMAN, L. 2004. Consistency for a simple model of random forests. 670, University of California, Berkeley, Berkeley.

CHI, M., JORDAN, P., VANLEHN, K. and HALL, M. 2008. Reinforcement Learning-based Feature Selection For Developing Pedagogically Effective Tutorial Dialogue Tactics, Proceedings of the 1st International Conference on Educational Data Mining, Montreal, 30-36.

CHI, M., VANLEHN, K. and LITMAN, D. 2010. Do Micro-Level Tutorial Decisions Matter: Applying Reinforcement Learning to Induce Pedagogical Tutorial Tactics, Intelligent Tutoring Systems. Springer Berlin / Heidelberg, Berlin, 224-234.

DUONG, M. and MOSTOW, J. 2010. Adapting a Prosodic Synthesis Model to Score Children's Oral Reading. In *Interspeech 2010*, Makuhari, Japan, Sept. 26-30, 2010.

EISENSTEIN, J., BARZILAY, R. and DAVIS, R. 2008. Gestural cohesion for topic segmentation. *ACL: HLT*, 852-860.

GONZÁLEZ-BRENES, J.P., BLACK, A.W. and ESKENAZI, M. 2009. Describing Spoken Dialogue Systems Differences. In *International Workshop on Spoken Dialogue Systems*, Irsee, Germany, 2009, Best Paper nominee.

GONZÁLEZ-BRENES, J.P. and MOSTOW, J. 2010. Predicting Task Completion from Rich but Scarce Data. In *Proceedings of the 3rd International Conference on Educational Data Mining*, Pittsburgh, PA, June 11-13, 2010, R.S.J.D. BAKER, A. MERCERON and P.I.J. PAVLIK, Eds., 291-292.

GONZÁLEZ-BRENES, J.P. and MOSTOW, J. 2011. Classifying Dialogue in High-Dimensional Space. *ACM Transactions on Speech and Language Processing 7*(3).

GUNAWARDANA, A., MAHAJAN, M., ACERO, A. and PLATT, J.C. 2005. Hidden conditional random fields for phone classification. In *9th European Conference on Speech Communication and Technology - Interspeech*

Lisboa, Portugal, 2005.

HAJDINJAK, M. and MIHELIC, F. 2006. The PARADISE evaluation framework: Issues and findings. *Computational Linguistics 32*(2), 263-272.

HUANG, X., ALLEVA, F., HON, H.W., HWANG, M.Y., LEE, K.F. and ROSENFELD, R. 1993. The SPHINX-II speech recognition system: an overview. *Computer Speech & Language 7*(2), 137-148.

LODHI, H., SAUNDERS, C., SHAWE-TAYLOR, J., CRISTIANINI, N. and WATKINS, C. 2002. Text classification using string kernels. *J. Mach. Learn. Res. 2*, 419-444.

LOURADOUR, J., DAOUDI, K. and BACH, F. 2006. SVM Speaker Verification using an Incomplete Cholesky Decomposition Sequence Kernel. In *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*, 28-30 June 2006, 2006, 1-5.

MÖLLER, S., ENGELBRECHT, K.P. and SCHLEICHER, R. 2008. Predicting the quality and usability of spoken dialogue services. *Speech Communication 50*(8-9), 730-744.

MÖLLER, S., SMEELE, P., BOLAND, H. and KREBBER, J. 2007. Evaluating spoken dialogue systems according to de-facto standards: A case study. *Computer Speech and Language 21*(1), 26 - 53.

MOSTOW, J. and AIST, G. 2001. Evaluating tutors that listen: An overview of Project LISTEN. In *Smart Machines in Education*, K. FORBUS and P. FELTOVICH, Eds. MIT/AAAI Press, Menlo Park, CA, 169-234.

MURPHY, K. 2012, in preparation. *Machine Learning: a Probabilistic Perspective*. MIT Press.

NG, A.Y. 2004. Feature selection, ℓ_1 vs. ℓ_2 regularization, and rotational invariance, ICML '04: Proceedings of the Twenty-first International Conference on Machine learning. ACM, Banff, Alberta, Canada, 78-86.

OBOZINSKI, G., TASKAR, B. and JORDAN, M.I. 2007. Multi-task feature selection, The Workshop of Structural Knowledge Transfer for Machine Learning in the 23rd International Conference on Machine Learning (ICML), Pittsburgh, PA.

QUATTONI, A., WANG, S., MORENCY, L.P., COLLINS, M. and DARRELL, T. 2007. Hidden conditional random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*(10), 1848-1852.

RABINER, L.R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*(2), 257-286.

REIGELUTH, C.M.A.S., F. S. (Editor), 1983. *The Elaboration Theory of Instruction*. Instructional Design Theories and Models: An Overview of their Current States. Lawrence Erlbaum, Hillsdale, NJ.

SINGH, S.P., KEARNS, M.J., LITMAN, D.J. and WALKER, M.A. 1999. Reinforcement Learning for Spoken Dialogue Systems, Advances in Neural Information Processing Systems. MIT Press, Cambridge, MA, 956-962.

STOLCKE, A., RIES, K., COCCARO, N., SHRIBERG, E., BATES, R., JURAFSKY, D., TAYLOR, P., MARTIN, R., ESS-DYKEMA, C.V. and METEER, M. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics 26*(3), 339-373.

SY BOR, W. 2006. Hidden Conditional Random Fields for Gesture Recognition, 2006, Q. ARIADNA, M. LOUIS-PHILIPPE, D. DAVID and D. TREVOR, Eds., 1521-1527.

WALKER, M., KAMM, C. and LITMAN, D. 2001. Towards developing general models of usability with PARADISE. *Natural Language Engineering 6*(3), 363-377.

WALLACH, H.M. 2006. Topic modeling: beyond bag-of-words, Proceedings of the 23rd international conference on Machine learning. ACM, Pittsburgh, Pennsylvania, 977-984.

XING, Z., PEI, J. and KEOGH, E. 2010. A brief survey on sequence classification. *SIGKDD Explor. Newsl. 12*(1), 40-48.

ZOU, H. and HASTIE, T. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 301-320.