Analyzing Dialog Coherence using Transition Patterns in Lexical and Semantic Features

Amruta Purandare and **Diane Litman**
Intelligent Systems Program
University of Pittsburgh
{amruta,litman}@cs.pitt.edu

Abstract

In this paper, we present methods to analyze dialog coherence that help us to automatically distinguish between coherent and incoherent conversations. We build a machine learning classifier using local transition patterns that span over adjacent dialog turns and encode lexical as well as semantic information in dialogs. We evaluate our algorithm on the Switchboard dialog corpus by treating original Switchboard dialogs as our coherent (positive) examples. Incoherent (negative) examples are created by randomly shuffling turns from these Switchboard dialogs. Results are very promising with the accuracy of 89% (over 50% baseline) when incoherent dialogs show both random order as well as random content (topics), and 68% when incoherent dialogs are random ordered but on-topic. We also present experiments on a newspaper text corpus and compare our findings on the two datasets.

Introduction

The field of discourse coherence has grown substantially over the past few years, from theories (Mann & Thompson 1988; Grosz, Joshi, & Weinstein 1995) to statistical models (Soricut & Marcu 2006; Barzilay & Lapata 2005; Lapata & Barzilay 2005) as well as to applications such as generation (Scott & de Souza 1990; Kibble & Power 2004), summarization (Barzilay, Elhadad, & McKeown 2002) and automatic scoring of student essays (Higgins *et al.* 2004). Most of these studies, however, have been conducted and evaluated on text datasets. Coherence is also important when it comes to speech and dialog based applications, so that a dialog system is able to make coherent conversations with users or detect places exhibiting a lack of coherence. For instance, (Stent, Prasad, & Walker 2004) use RST-based coherence relations for dialog generation. Other studies on dialogs (Rotaru & Litman 2006) and spoken monologues (Passonneau & Litman 1993; Nakatani, Hirschberg, & Grosz 1995) have primarily looked at the intentional structure of discourse (Grosz & Sidner 1986) rather than the informational structure that is captured by recent statistical models of coherence. In this paper, we apply and extend these statistical models of text coherence (Marcu & Echihabi 2002; Lapata & Barzilay 2005) and information ordering (Lapata 2003) to dialogs such that a dialog system can automatically distinguish between coherent and incoherent conversations.

Consider the following two dialogs:

A: Have you seen Dancing with Wolves?
B: Yeah, I've seen that. That was a really good movie. Probably one of the best things about it was the scenery.
A: I thought the story was pretty good too. I think Kevin Costner did a really good job with it.
B: Have you ever lived in that part of the country?
A: No I haven't.

Figure 1: Coherent Dialog

A: So, what do you think are the major causes of air pollution?
B: I uh enjoy Szechuan type of Chinese food.
A: That's great! So do you still sing?
B: Yeah I do, I have a seven and half year old dog.
A: I had a Chevrolet before I bought my Taurus.
B: I think, we can spend our money better elsewhere.

Figure 2: Incoherent Dialog

While the first dialog illustrates a fluent, coherent conversation[1], the second one is just a random collection of utterances[2] with no connection to each other. Our objective in this paper is to design an algorithm that can automatically tell if a given dialog is coherent or not.

(Barzilay & Lapata 2005) model text coherence as a ranking or ordering problem by finding the most acceptable order of given n sentences. Here, we instead formulate coherence assessment as a binary classification task in which our goal is to simply label dialogs as coherent or incoherent. This framework is particularly suitable for

[1]This example is taken from the Switchboard dialog corpus

[2]These turns are randomly selected from different Switchboard dialogs

the evaluation of dialog generation (Walker *et al.* 2004; Higashinaka, Prasad, & Walker 2006; Chambers & Allen 2004) and simulation models (Schatzmann, Georgila, & Young 2005) that aim towards generating natural and coherent dialogs almost indistinguishable from human-human conversations (Ai & Litman 2006).

The paper is organized as follows: We first discuss our data collection and how easily we create a corpus of coherent and incoherent dialogs. We then describe our features and feature selection strategy. We then present and discuss our results on the Switchboard dialog corpus. We perform similar experiments on a newspaper text corpus, compare our findings on the two datasets (text and dialogs), and finally end with a summary of conclusions.

Dialog Corpus

For our experiments, we need a corpus that represents examples of both coherent and incoherent dialogs. Following the work on information ordering (Lapata 2003; Soricut & Marcu 2006) that uses the original sentence order in the document as the reference for comparison, we use original dialogs as seen in some *real-corpus* as our coherent examples. Thus, we use the term *coherence* somewhat loosely here for naturally-ordered, real human-human dialogs.

For these experiments, we used dialogs from the Switchboard corpus (Godfrey & Holliman 1993). This corpus contains a total of 2438 dialogs (about 250,000 dialog turns and 3M words). Each dialog is a spontaneous telephone conversation between two speakers who are randomly assigned a topic from a set of 70 topics. There are 543 speakers in total and the topic/speaker assignment is done such that no speaker speaks on the same topic more than once and no two speakers get to converse together more than once. This gives us a set of 2438 coherent dialogs.

Incoherent examples are then created automatically using two types of shuffling methods that are described below:

Hard Shuffle: For each Switchboard dialog, we create a corresponding incoherent example by randomly shuffling its dialog turns. As the turns from each dialog are shuffled separately, the corresponding incoherent version has the same overall content as the original dialog, but in random order. Because the original Switchboard dialogs are on one topic, the incoherent dialogs thus created are also on a single topic. This gives us a collection of 2438 incoherent dialogs (by considering only one possible random order for each Switchboard dialog) that has the same total number of turns and words as the coherent set.

Easy Shuffle: We also create a second incoherent dialog set by randomly shuffling turns from all Switchboard dialogs together. These incoherent examples are, thus, not guaranteed to be on a single topic. Specifically, these dialogs not only have a random order but also random content (topics). For this shuffling, we treated end-of-dialog boundaries as if they are regular dialog turns, so that the shuffling program automatically inserts dialog end boundaries. This also gives us a total of 2438 incoherent dialogs that have the same total number of turns and words as the original coherent set as well as the other incoherent set.

Using the above two shuffling methods, we then create two datasets which we refer to as Switch-Easy and Switch-Hard, each containing a total of 4876 dialogs of which 2438 (50%) are coherent (original Switchboard) and 2438 (50%) are incoherent (random-order) created using either Easy or Hard shuffle. We expect that the algorithm we build to distinguish between coherent and incoherent dialogs will perform better on the Switch-Easy set than on Switch-Hard as the Easy dialogs not only present random order but also random topics.

Caveats: While the above procedure offers the nice advantage of automatically creating a large corpus of coherent and incoherent dialogs without any manual annotations, we expect and realize that not all dialogs in the real-corpus (like Switchboard) will be coherent; neither will all random-order examples created by shuffling be completely incoherent. Our future studies will explore methods for identifying such outliers.

Features

Coherence being a discourse-level phenomena, we need features that span over and model relations between multiple dialog turns. The features we use here are borrowed from the previous work on text structuring (Lapata 2003) and recognizing discourse relations (Marcu & Echihabi 2002). First, each dialog turn is represented by a set of features. Then, from each pair of adjacent dialog turns, we extract transition patterns by taking the cross-product of their feature sets. For example, if T_i and T_{i+1} are two adjacent dialog turns such that T_i has 3 features $\{f_1, f_2, f_3\}$ and T_{i+1} has 2 features $\{f_4, f_5\}$, then the method will extract six transition patterns: $\{f_1\text{-}f_4, f_1\text{-}f_5, f_2\text{-}f_4, f_2\text{-}f_5, f_3\text{-}f_4, f_3\text{-}f_5\}$ from this pair of dialog turns. In general, given a sequence of k consecutive dialog turns T_i - T_{i+1} - T_{i+2} - ... - T_{i+k-1}, a transition pattern shows a sequence of k features f_0-f_1-f_2-...-f_{k-1} taken from the cross-product of their feature sets, i.e. $f_0 \epsilon T_i, f_1 \epsilon T_{i+1}, f_2 \epsilon T_{i+2}$ and so on. The total number of patterns extracted from k consecutive turns is thus the product of the cardinalities of their feature sets. Due to time and computational constraints, we currently analyze only *local* transition patterns from adjacent dialog turns.

In this paper, we create transition patterns using two types of features:

Lexical: Each dialog turn is represented as a feature set of words that appear in the turn (removing common stopwords). A lexical transition pattern w_1-w_2 is a pair of words such that words w_1 and w_2 appear in adjacent dialog turns. The frequency of the pattern w_1-w_2 in the corpus counts how often word w_1 in the present turn is followed by word w_2 in the next turn, or the number of adjacent dialog turns that demonstrate a transition pattern w_1-w_2. Interestingly, we noticed that some of the most frequent lexical patterns in our data are those for which $w_1 = w_2$, e.g. *hi-hi, bye-bye, school-school, tax-tax, music-music, read-read* etc, which suggests that adjacent turns in our dialogs often show the same lexical content.

Semantic: These features are used in order to capture coherence at the semantic level, without relying on surface level lexical matchings. While (Lapata & Barzilay 2005)

use Latent Semantic Analysis and Wordnet-based similarity metrics for their semantic model, we here use a simple and efficient technique to analyze semantic coherence by using corpus-derived semantic classes of words created by the CBC (Clustering By Committee) algorithm (Lin & Pantel 2002). The output of CBC shows clusters of distributionally similar words such as *N719: (Honda, Toyota, Mercedes, BMW, ...)*, *N860: (bread, cake, pastry, cookie, soup ...)*, *N951: (onion, potato, tomato, spinach, carrot ...)* etc, where *N719, N860, N951* show their cluster ids. There are 2211 clusters of over 50,000 words in CBC generated from 1GB corpus of newspaper text (Lin & Pantel 2002).

For each lexical pattern w_1-w_2, we create a corresponding semantic pattern c_1-c_2 by simply replacing each word by its CBC cluster id. As a result, lexical patterns whose corresponding words are semantically similar (belong to the same CBC cluster) map to the same semantic pattern. For example, here, lexical patterns *carrot-cake, potato-bread, tomato-soup* will map to the same semantic pattern *N951-N860*. In cases where a word maps to multiple clusters (that represent its multiple senses), we currently create a semantic pattern for each cluster that it belongs to. In the future, we will incorporate methods to disambiguate words based on their contexts.

Feature Selection

We extract transition patterns from both positive (coherent) and negative (incoherent) examples so that we do not use the actual class labels at the time of feature selection (prior to training). While we could simply use all transition patterns in the data as our features, there are over 4M lexical and 700K semantic patterns in each Switch-Easy and Switch-Hard dataset. Not only it is challenging to process and classify data in such a high dimensional feature space, but features that occur rarely are also not very helpful in making a coarse-level binary distinction. To address this, we score patterns using the log-likelihood ratio (Dunning 1993) and retain only those patterns that show significant dependencies ($p < 0.01$ or log-likelihood score $>= 6.64$). This rejects the null hypothesis of independence with 99% confidence and gets rid of a lot of rare insignificant patterns that occur by chance. As incoherent dialogs are randomly ordered, we expect that most patterns observed in incoherent examples won't repeat often, neither in coherent nor in other incoherent examples (as they are also randomly ordered). In other words, incoherent dialogs will exhibit random transition patterns that we expect will get filtered out by the log-likelihood test. On the other hand, most significantly recurring patterns will primarily appear in coherent dialogs. Thus, this feature selection strategy indirectly identifies features that characterize coherence without using their true class labels. After applying the log-likelihood filter, we obtained approximately 500K lexical and 30K semantic patterns for each of the Switch-Easy and Switch-Hard datasets.

Experiments

We model the task of identifying coherent and incoherent dialogs as a binary classification problem in which the algorithm is presented with examples from the two classes and is asked to classify dialogs as coherent or incoherent. For this, we represent each dialog example as a feature vector whose dimensions are transition patterns and their corresponding feature values indicate the number of adjacent turns in the given dialog that exhibit a certain transition pattern. In other words, a feature vector is created per dialog, by counting the occurrences of each pattern over each pair of adjacent turns in that dialog.

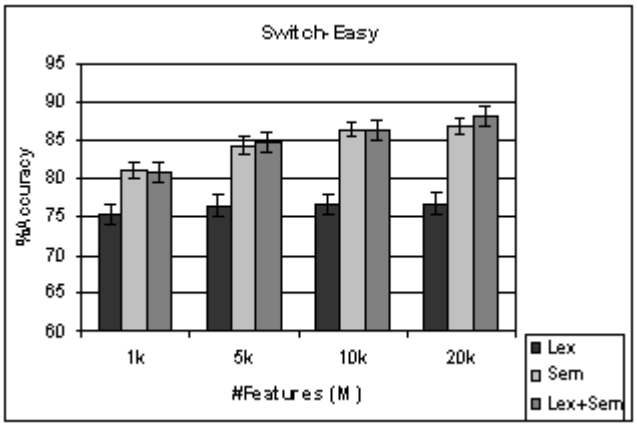

Figure 3: Results on Switch-Easy Dataset

We run a 10-fold cross validation experiment using the Naive Bayes classifier from the Weka toolkit. We conduct experiments using lexical and semantic patterns, used separately as well as together. We also experiment with different sizes of feature sets by selecting only the top M most significant patterns for M = [1K, 5K, 10K, 20K]. For the Lexical + Semantic combination, we use half lexical and half semantic patterns. For example, for a feature set of size 10K, there are exactly 5K lexical and 5K semantic patterns. In the future, we plan to create feature sets based on their significance levels (p-values).

Figures 3 and 4 show the performance of the classifier (% accuracy) plotted against different sizes of feature sets, for Lexical, Semantic and Lexical + Semantic features on Switch-Easy and Switch-Hard datasets respectively. Small vertical bars indicate the confidence intervals, computed as $mean \pm (2 * standard - error)$ over 10 folds. Results with non-overlapping confidence intervals are statistically different with 95% confidence. As these figures show, all results are significantly above the 50% random baseline[3] with the accuracy numbers ranging from 75% to almost 90% on Switch-Easy and about 62-68% on Switch-Hard.

On Switch-Easy set (see figure 3), we notice that semantic features perform much better than lexical and that there is no advantage to combining lexical and semantic features together over semantic features alone. We can also notice that the performance of semantic and lexical + semantic features boosts up from 80% to 89% when the feature set size

[3]Distribution of coherent and incoherent dialogs is equal (50-50) for each dataset used in this paper.

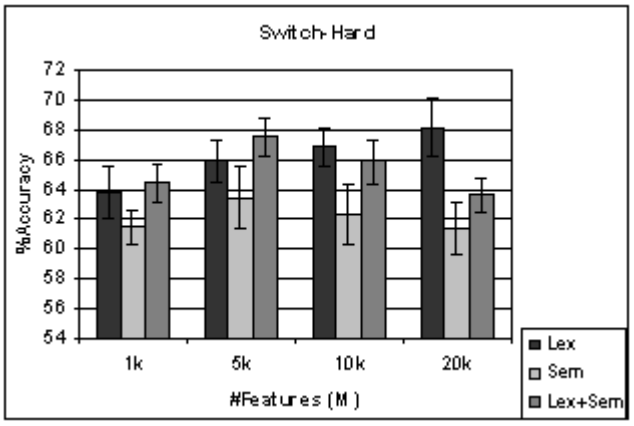

Figure 4: Results on Switch-Hard Dataset

is increased from 1K to 20K. Lexical features, on the other hand, do not show a significant improvement in the accuracy with additional features. Thus, when incoherent dialogs have both random order and random content, a classifier can discriminate among coherent and incoherent dialogs with a very high accuracy (almost 90%) using semantic patterns and about 76% using lexical patterns.

Results on Switch-Hard dataset (refer to figure 4) are, as expected, much lower than on Switch-Easy, although still significantly above the 50% baseline. In contrast to what we noticed on Switch-Easy dataset, here, semantic features do not perform as well as the other two. Interestingly, lexical features show a consistent improvement in the accuracy with more features, whereas, the performance of lexical + semantic features first improves but then degrades when M is increased beyond 5K. The overlapping confidence intervals, however, show that most of these differences are not statistically significant. In summary, when incoherent dialogs are random-ordered but on-topic, a classifier can distinguish between coherent and incoherent dialogs with the best accuracy of about 68% using 5K lexical + semantic patterns or 20K lexical patterns.

The reason we think that semantic features perform better on Switch-Easy but not so well on Switch-Hard is as follows: semantic features use abstract semantic classes of words that group similar words. Since incoherent examples created using Easy shuffle show topically unrelated content, transition patterns occurring in these examples also contain semantically unrelated words. Patterns present in coherent examples, on the other side, will have semantically related words. By mapping words to their semantic classes, semantic features allow us to capture these coarse-level topic distinctions for Easy examples. For the Hard dataset, both coherent and incoherent dialogs are on-topic and hence both examples will show transition patterns of semantically related words. Thus, mapping words to their abstract semantic classes does not offer any advantage to distinguish between two sets of dialogs that are both on-topic and contain semantically related content.

Experiments on a Text Corpus

Spontaneous spoken conversations as found in the Switchboard dialog corpus generally tend to be less coherent than formal written text. We, therefore, expect that our algorithm should perform even better on a text corpus than it did on the dialog corpus. In this section, we test this hypothesis by conducting similar experiments on a newspaper text corpus. As the Switchboard corpus is relatively small in size (compared to available text corpora), to be fair, we created a text corpus of comparable size by randomly selecting 2500 news stories (documents) from the Associated Press (AP) newswire text. Thus, the number of dialogs in Switchboard (2438) matches approximately the number of documents (2500) in the selected text corpus.

While we attempt to make a fair comparison between the two experiments here, there is, however, one issue that we would like to point out: although our text and dialog datasets match in the number of documents = dialogs, text data has much smaller number of words (900K) in comparison to Switchboard (3M). Also, the number of sentences in the selected AP text (46K) does not match with the number of dialog turns in Switchboard (250K). When we attempted to create a text corpus that matches with Switchboard in terms of the number of words or sentences = turns, it offered different number of documents. In short, we found it very hard to create a text corpus that matches with our dialog corpus in all of the parameters (such as the number of words, sentences, documents etc). Here, we choose to fix the number of text documents to match the number of dialogs because when it finally comes to classification, the accuracy of a machine learning algorithm primarily depends on the number of instances (here, documents or dialogs) and the number of features (which we control by selecting the top M most significant patterns). Other factors (such as the number of words, sentences etc) are mostly hidden from the classifier although they may indirectly influence the sparsity of data representation.

The text corpus we use for these experiments, thus, consists of 2500 documents collected from the AP newswire corpus. Sentence boundaries are detected automatically using the sentence boundary detection tool from (Reynar & Ratnaparkhi 1997). Similar to dialog experiments, these original documents are treated as coherent text samples. Incoherent examples are created in the same manner using Easy and Hard shuffling methods described earlier. In short, incoherent texts produced by Hard shuffle contain sentences from the same original document but only in random order, whereas, Easy shuffle creates incoherent texts that contain sentences randomly selected from different documents. This gives us two text datasets to experiment with: AP-Easy and AP-Hard, each of which contains a total of 5000 documents with 2500 coherent (original AP) and 2500 incoherent (produced either by Easy or Hard shuffle).

Feature extraction and selection is done in the same manner as that for the dialog corpus by treating each sentence as one turn and extracting transition patterns from pairs of adjacent sentences. Figures 5 and 6 show results of the 10-fold cross validation experiment on AP-Easy and AP-Hard datasets conducted under the same settings as that for the

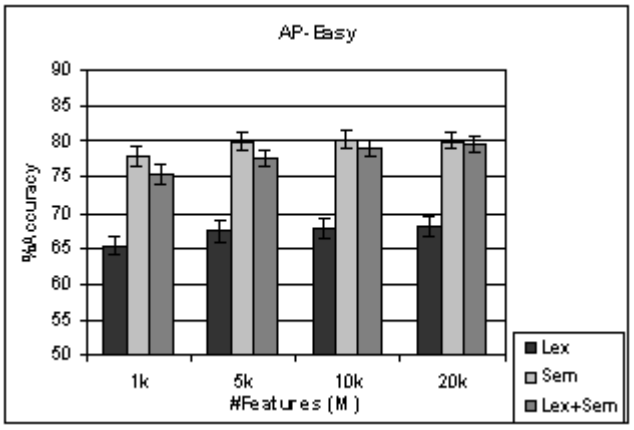

Figure 5: Results on AP-Easy Dataset

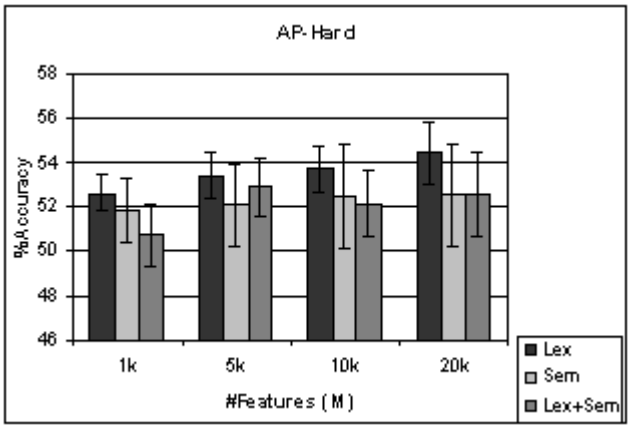

Figure 6: Results on AP-Hard Dataset

dialog corpus.

Similar to Switch-Easy, AP-Easy also shows better performance with semantic features than with lexical, and no improvement on combining lexical and semantic features together over semantic features alone. On Switch-Easy, we had noticed that the accuracy of semantic and lexical + semantic features was significantly improved (from 80% to almost 90%) on adding more patterns. But on AP-Easy, the performance for all three features improves only slightly (by 3-5%) when M is increased from 1K to 20K.

Results are quite poor on AP-Hard dataset (see figure 6) with accuracies of 51-54%. This might suggest that the problem of distinguishing between coherent and incoherent texts is much harder when incoherent texts are created by shuffling sentences from the same original document (on-topic). This also suggests that even after shuffling, the two partitions (original coherent and shuffled incoherent) are still highly similar and show similar local transition patterns. On the other hand, for the dialog dataset (Switch-Hard), we saw that even when incoherent dialogs were on-topic, the classifier could still distinguish between coherent and incoherent dialogs with a fairly decent accuracy (about 62-68%).

Thus, while we expected that results would actually be better on the text corpus than on Switchboard dialogs, to our surprise, we notice the opposite. On AP-Easy, the best result is about 80% (compared to 89% on Switch-Easy), whereas on AP-Hard, figures are mostly in low 50s (compared to 68% on Switch-Hard). The reason could be that formal written text as in newspaper articles often shows very rich vocabulary and word-usage, whereas, spontaneous spoken dialogs, where speakers think about the content offhand will have more repetitions. A quick look at the data indeed shows that the text collection has a higher types/tokens ratio (33%) compared to Switchboard (10%), although the number of words (tokens) is higher for Switchboard (3M) than for text (900K). The other reason could be that although our text corpus matches Switchboard in the number of instances (documents = dialogs), these documents are much shorter in length compared to Switchboard dialogs (in terms of the number of words or sentences). This makes it even harder for the classifier as there are fewer features per example.

Conclusions

In this paper, we presented a simple framework that automatically classifies dialogs as coherent or incoherent. Our coherent examples are real human-human dialogs in their original order taken from the Switchboard corpus, whereas incoherent examples are random-ordered dialogs created using two shuffling methods. While the first method ensures that incoherent examples are on a single topic, the second method produces incoherent dialogs that not only show random order but also random topics. From these examples, we learn transition patterns in lexical and semantic features that span over adjacent dialog turns. These patterns are then supplied as features to a machine learning classifier that automatically labels dialogs as coherent or incoherent.

Our results show that when incoherent dialogs have random order as well as random content, semantic features perform much better than lexical, with the best accuracy of about 89% for semantic features compared to 76% for lexical. Results are lower and in the range of 62-68% when incoherent dialogs are randomly ordered but on-topic. On these examples, we see that semantic features do not perform as well as lexical. We provide a reasoning that since semantic features map words to abstract semantic classes, they allow us to capture coarse-level topic distinctions in order to separate on-topic coherent dialogs from random topic incoherent dialogs. When both coherent and incoherent dialogs are on-topic, mapping words to their semantic classes is not very useful.

We also presented results on a newspaper text corpus that has a comparable size to our dialog corpus. We showed that while some of the findings generalized to both text and dialog corpora, others did not. Specifically, on this dataset also, semantic features work better when incoherent examples have random content. To our surprise, we found that results are much lower on the text corpus compared to the dialog corpus, especially when both coherent and incoherent texts are on-topic. We hypothesize that although written text generally tends to be more coherent than spontaneous

spoken dialogs, rich vocabulary and word usage in formal written text also makes the problem more challenging.

In the future, instead of labeling entire dialogs as coherent or incoherent, we would like to perform a more fine-grained analysis and specifically identify coherent and incoherent parts within each dialog. This will hopefully address some of the caveats we mentioned earlier in the paper that real human dialogs are not always completely coherent; neither all random-order dialogs are completely incoherent. We also plan to conduct a similar study on acted (or portrayed) dialogs such as from movies and tv-shows, and see how the results compare with our current results on spontaneous Switchboard dialogs.

Acknowledgments

Authors would like to thank the anonymous reviewers and members of the ITSPOKE group for their insightful comments and feedback.

References

Ai, H., and Litman, D. 2006. Comparing real-real, simulated-simulated, and simulated-real spoken dialogue corpora. In *Proceedings of the AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*.

Barzilay, R., and Lapata, M. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the Association for Computational Linguistics (ACL)*, 141–148.

Barzilay, R.; Elhadad, N.; and McKeown, K. 2002. Inferring strategies for sentence ordering in multidocument summarization. *Journal of Artificial Intelligence* 17:35–55.

Chambers, M., and Allen, J. 2004. Stochastic language generation in a dialogue system: Toward a domain independent generator. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, 9–18.

Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19(1):61–74.

Godfrey, J., and Holliman, E. 1993. *Switchboard-1 Transcripts*. Linguistic Data Consortium, Philadelphia.

Grosz, B., and Sidner, C. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics* 12(3).

Grosz, B.; Joshi, A.; and Weinstein, S. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics* 2(21):203–225.

Higashinaka, R.; Prasad, R.; and Walker, M. 2006. Learning to generate naturalistic utterances using reviews in spoken dialogue systems. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Higgins, D.; Burstein, J.; Marcu, D.; and Gentile, C. 2004. Evaluating multiple aspects of coherence in student essays. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 185–192.

Kibble, R., and Power, R. 2004. Optimizing referential coherence in text generation. *Computational Linguistics* 30(4):401–416.

Lapata, M., and Barzilay, R. 2005. Automatic evaluation of text coherence: Models and representations. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, 1085–1090.

Lapata, M. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the Association for Computational Linguistics (ACL)*, 545–552.

Lin, D., and Pantel, P. 2002. Concept discovery from text. In *Proceedings of Conference on Computational Linguistics (COLING)*, 577–583.

Mann, W., and Thompson, S. 1988. Rhetorical structure theory: Rhetorical structure theory: Toward a functional theory of text organization. *Text* 8(3):243–281.

Marcu, D., and Echihabi, A. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Nakatani, C.; Hirschberg, J.; and Grosz, B. 1995. Discourse structure in spoken language: Studies on speech corpora. In *Working Notes of the AAAI-95 Spring Symposium in Palo Alto, CA, on Empirical Methods in Discourse Interpretation*, 106–112.

Passonneau, R., and Litman, J. 1993. Intention-based segmentation: Human reliability and correlation with linguistic cues. In *Proceedings of the Association for Computational Linguistics (ACL)*, 148–155.

Reynar, J., and Ratnaparkhi, A. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*.

Rotaru, M., and Litman, D. 2006. Exploiting discourse structure for spoken dialogue performance analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Schatzmann, J.; Georgila, K.; and Young, S. 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, 45–54.

Scott, D., and de Souza, C. 1990. Getting the message across in RST-based text generation. In *Current research in natural language generation*, 47–73.

Soricut, R., and Marcu, D. 2006. Discourse generation using utility-trained coherence models. In *Proceedings of the Association for Computational Linguistics (ACL)*, 803–810.

Stent, A.; Prasad, R.; and Walker, M. 2004. Trainable sentence planning for complex information presentations in spoken dialog systems. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL)*, 79–86.

Walker, M.; Whittaker, S.; Stent, A.; Maloor, P.; Moore, D.; Johnston, M.; and Vasireddy, G. 2004. Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science* 28(5):811–840.