Human Computation and Serious Games: Papers from the 2012 AIIDE Joint Workshop
AAAI Technical Report WS-12-17

Modeling Topics in User Dialog for Interactive Tablet Media

Adrian Boteanu, Sonia Chernova
Worcester Polytechnic Institute
aboteanu@wpi.edu, soniac@wpi.edu

Abstract

In this paper, we present a set of crowdsourcing and data processing techniques for annotating, segmenting and analyzing spoken dialog data to track topics of discussion between multiple users. Specifically, our system records the dialog between the parent and child as they interact with a reading game on a tablet, crowdsources the audio data to obtain transcribed text, and models topics of discussion from speech transcription using ConceptNet, a freely available commonsense knowledge base. We present preliminary results evaluating our technique using dialog collected using an interactive reading game for children 3-5 years of age. We successfully demonstrate the ability to form discussion topics by grouping words with similar meaning. The presented approach is entirely domain independent and in future work can be applied to a broad range of interactive entertainment applications, such as mobile devices, tablets and games.

Introduction

The ability to model and accurately respond to user behavior lies at the heart of all interactive entertainment technologies. While in traditional interactive media user input has largely been limited to button or mouse presses, recent broad adoption of sensor-rich devices, such as tablets and the Microsoft Kinect, have greatly expanded the range of available inputs. With today’s devices, developers seeking to model user activity have access to touch, gesture, accelerometer, audio and 3D posture data in addition to traditional inputs.

Speech plays a particularly interesting role in modeling user interactions. Speech recognition is a notoriously difficult problem, resulting in few technologies that rely on that form of input. However, understanding what the user is saying, particularly understanding dialog between multiple users in the context of social applications, can greatly increase the ‘awareness’ of a program and its functionality. For example, tracking the topics of discussion that come up as a parent and child play a game or read a book together can enable the application to customize its content to the user's interests.

In this paper, we examine a particular application for dialog modeling – the development of an interactive reading primer for tablets that seeks to foster early literacy skills and shared parent-child reading for children 3-5 years of age. Children in this age group go through a rapid learning period, during which it is particularly important for them to have interactions with adults, especially their parents, from which they can learn. Our project seeks to foster greater parent-child engagement through the use of a targeted topic suggestion system that:

1) records the dialog between the parent and child as they interact with a reading game on a tablet
2) crowdsources the audio data to obtain transcribed text
3) extracts topics of discussion from speech transcription
4) crowdsources topics of discussion from across a large population of readers
5) provides discussion topic suggestions to individual readers

The ultimate goal of this project is to record the interactions and discussions of parent-child pairs across a community of readers, leverage this information, in combination with a common sense knowledge base, to develop computational models of the interactions, and use these models to provide context-sensitive discussion topic suggestions to parents during the shared reading activity.

In this paper, we present our progress through the first three stages outlined above. We report lessons learned in using Amazon’s Mechanical Turk to annotate dialog data between young children and their parents, as well as preliminary results in modeling discussion topics. While our study focuses specifically on a reading app for Android tablets, we believe that the models and results obtained through this work will generalize to a broad range of interactive media applications, such as games and mobile devices.

Related work

Significant work has been done in tracking topics in news, scientific literature and meetings by using data mining methods and machine learning (Krause, Leskovec, Guestrin 2006; Eisenstein, Barzilay 2008). However, in that context, topics have a different meaning, focusing on particular events or well defined problems. Common methods used for analysis include finding frequent words of interest and using data mining methods to predict their field. Unlike the casual and ambiguous discussions that we analyze in this project, these methods rely on precise definitions of keywords and historic events. One key difference in our work is that words with multiple meanings can be part of multiple topics, and the tracking process follows each particular meaning separately.

Among entertainment applications, such as games, dialog modeling has also been explored by Reckman et al. (2010, 2011). In their work, the authors use virtually grounded dialogue data from a virtual world game to automatically learn words, grammatical constructions and their meanings. Our current work focuses only on speech data, but our aim is to expand our techniques to also incorporate visual cues in the future by analyzing focus of attention from touch and gesture data on the tablet. This may bring our work closer to that of Reckman et al. in its ability to leverage visually grounded information.

Finally, in order to understand how different topics relate to each other, it is important to understand the meaning or concept behind each word. ConceptNet is a freely available commonsense knowledge base and natural-language-processing tool-kit which supports many practical textual-reasoning tasks over real-world documents, including topic-gisting, analogy-making, and other context oriented inferences. (Havasi, Speer, Alonso 2009). From ConceptNet we use a singular value decomposition matrix of the graph relations to measure the pairwise similarity distance between words. In selecting the significant words, for this project we use a new method which takes into account the human short term memory for identifying words of interest that will be grouped into topics.

Data collection and annotation

Figure 1 presents an overview of the data flow in our system, divided into three stages:

- Stage 1: data collection and annotation
- Stage 2: finding concepts that users are interested in and grouping them into a general topic model
- Stage 3: tracking particular topics across sessions using the general model

This section describes the first stage, including audio data capture, preprocessing through segmentation and the annotation process. The following sections will describe the other two stages.

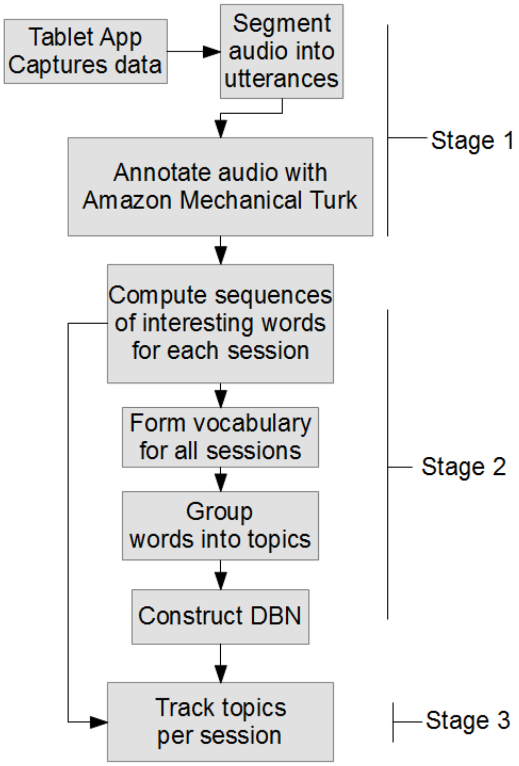
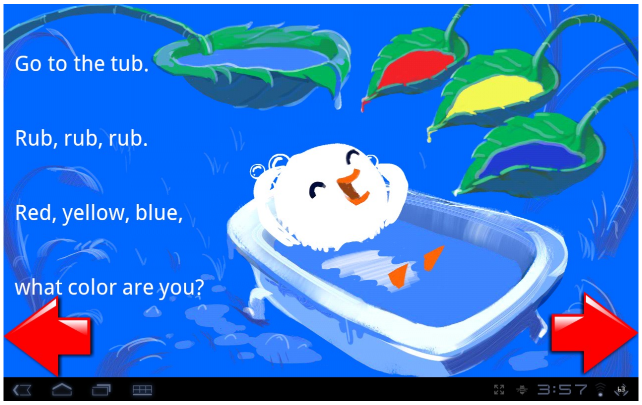

Tablet App Captures data
Segment audio into utterances
Annotate audio with Amazon Mechanical Turk
Stage 1
Compute sequences of interesting words for each session
Form vocabulary for all sessions
Group words into topics
Construct DBN
Stage 2
Track topics per session
Stage 3

Figure 1. General system diagram. Stage 1 refers to data capture and annotation. Stage 2 computes words of interest and topic groupings so that stage 3 can track topics in conversations.

Data capture

The Android application we use for collecting data is the TinkrBook interactive reading primer (Chang, Breazeal 2011), which follows a day in the life of a duck, going from bathing to eating and then going to sleep at night. As the story follows the duck through these tasks, it requires children to solve small challenges, such as counting or recognizing colors. Completing each task advances the story. Although the story contains clues for completing the puzzles, it is intended to be read by parents and children together.

The story application records audio with the onboard tablet microphone, touch gestures and what objects from the story the users manipulate. The readers interact with the story through tapping and dragging. For privacy concerns, recording can be disabled at any time (Figure 2).



Go to the tub.

Rub, rub, rub.

Red, yellow, blue,

what color are you?

Figure 2. In story mini-game, the task is to combine colors.

Once collected, the audio data forms a continuous stream. We use a tool from CMU Sphinx (Lamere et al. 2003) to split this stream into utterances that have at least one second pauses in between. This produces, with some margin of error, sentences spoken by a single person, sessions of quick dialogue with no pause and background noise. Common sources of background noise are the tablet hitting surfaces such as a table, TV sets playing and other people speaking unrelated to the tablet. The average length of each audio segment is 5 seconds, with the shortest being under one second and the longest at over 15 seconds. Each speech utterance contains ten words on average. One story reading produces on average 300 audio fragments out of a 20 minute long audio recording. The corpus is to be expanded. It currently consists of 5 hours of interaction, captured from 9 families out of which one hour was annotated.

Crowdsourcing data annotation

Amazon Mechanical Turk is a framework for crowdsourcing small tasks. The service version we used for this project involves hosting HTML/Javascript questionnaires on our server, which are included in tasks called HITs hosted on the Mechanical Turks servers. After listening to the audio clip, the workers classify the audio source as adult, child, both adult and child, or noise. If speech is present, workers enter the speech transcription into a text box.

We employ a number of techniques to enhance the transcription process. The first is to check all transcriptions against a general English dictionary for spelling correctness. We compared two strategies, one of immediately accepting the first response for each audio clip and a second one of iterative improvement. For the latter, each transcription is progressively improved, until the Levenshtein distance (Levenshtein 1965) between two consecutive transcriptions is smaller than a threshold value or until more than ten iterations have been made. Since there is a chance of workers returning unreliable transcriptions, we submit data in small batches of less than 30 audio clips at a time, so that fewer transcriptions are affected by one individual worker that might be working carelessly. We found that workers were less likely to start working on a small set of HITs. Therefore, the money reward is gradually increased as the number of HITs in the working set decreases so that they are more appealing to work on. This reduces the cost per audio fragment while increasing the return time (Figure 3).

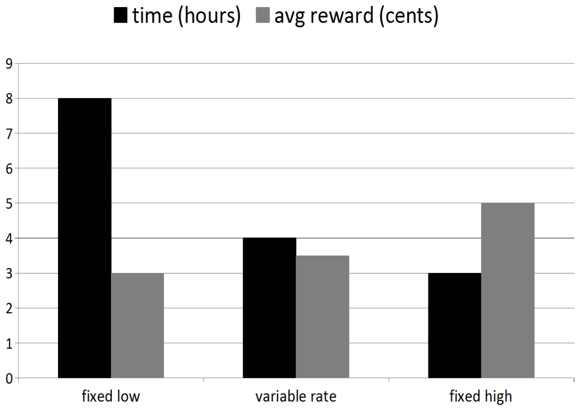


Figure 3. Average reward per task against average completion time for the same number of tasks. A low rewards are 3 cents/HIT, high rewards are 5 cents/HIT and the variable rate goes from 3 to 5 cents/HIT.

Discussion topic modeling

This section covers Stage 2 of the block diagram (Figure 1), describing how we use the annotations from multiple reading sessions (by the same family) to generate discussion topics. We then use these topics to construct a general inference network that can used for identifying the most likely topic at a given moment in other discussions.

Interest metric and language normalization

The first step in grouping words by topic is to identify which words are the most relevant for the readers as they are talking about the story. Written text and speech are very different in terms of phrasing, word selection and connectors. In particular, parents talking with their children have other goals beside transmitting a message, such as teaching new words (Hausendorf, Quasthoff 1992). We observed that parents often have to restate the goals or some important concepts to keep their children on track.

Another characteristic of spoken speech is that the density of topic-relevant words, such as nouns, verbs and adjectives, is low when compared to a written document.

Finally, the noise introduced by the Mechanical Turk transcription process itself provides an additional challenge, introducing errors due to misheard words, worker fatigue or laziness.

Because of these factors, along with the fact that the temporal location of words within a discussion is important for making relevant topic suggestions, we introduce a new metric for measuring how interested the user is in some words during dialogue. This is different from document categorization techniques that look for cues such as jargon, personalities and specific events. We track common words based on how significant they might be to the users for reading the story. One premise of the metric is that more frequently repeated words have greater importance. Our second premise is that the user's short term memory, which is used in dialogue, needs to have the concepts of interest constantly refreshed, especially in the case of children. This is achieved by either the child asking questions or by the adult repeating some words more than he/she would when discussing with an adult (Hausendorf, Quasthoff 1992).

We preprocess the text by removing all common words and particles, such as “is”, “the” and “it”. Then we use the Snowball stemmer (Porter, 2011) on all words. We use a set structure to capture all unique word stems and associate each with two numbers. The first is the interest metric value, *m*, and the second is the word’s relative frequency between its first and last occurrence, *f*. The system works in steps, one new word representing a new step, analyzing the words in the order in which they were spoken. At each step, the system updates the interest metric value for all words in the set, either increasing the interest metric value if the word occurred at the present step or decreasing it otherwise (Figure 4).

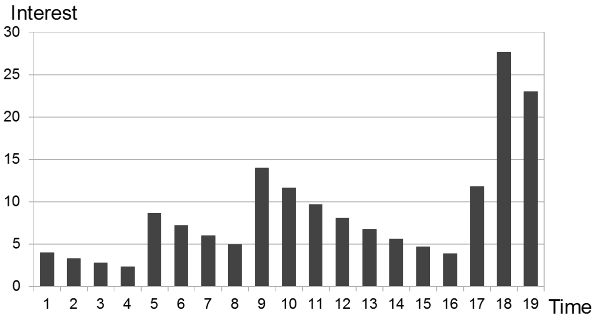


Figure 4. After occurring frequently at steps 1, 5 and 9, the word is not spoken for a longer time. Then, at steps 17 and 18 the interest metric value is increased accounting for previous occurrences.

The second value, the relative frequency, is used to model the speaker remembering a particular word. If at some point in the past a word was spoken frequently but it has not been spoken in a while, the interest metric value will be increased more (Figure 4). However, the relative frequency is then calculated over the entire span, from the first occurrence, so that one word does not become unnecessarily important (Figure 5). This enables elements that are present throughout the story, such as the protagonist, to be gradually replaced by newer and more local concepts, offering the reader varied suggestions that follow the story better. For the same purpose, we limit the maximum interest metric value a word can have.

```
1. if word occurred:
2.     if $m < lowMetricThr$ and $f > highFreqThr$:
3.         $m = m * \frac{a}{1-f}$
4.     $m = b * (m+c)$
5.     $f = updateFrequency(f)$
6. else:
7.     $m = \frac{m}{d} - e$
```

Figure 5. Basic algorithm for calculating the metric value. Lines 2 and 3 simulate the recent memories that the speakers would have about past discussions. m stands for the interest metric value, f for the relative word frequency. All other variables are adjustable parameters. For all the results presented here we used a = 2, b = 2. c = 2, d = 1.1 and e = 0.

After getting a new word and updating the interest metric values of the entire set, the system outputs the set of words that have the interest metric value higher than a given threshold. We consider these words to be the relevant, or interesting, words in the readers’ perception at that particular moment. The result is a sequence of (possibly empty) sets of words, one for each relevant stemmed word, along with a full set of all the words selected by the interest metric, the vocabulary of interest. This vocabulary is used for creating topics and the sequence of words for tracking topics.

Grouping words by topic

In this section, we describe the last two blocks from Stage 2 of Figure 1: how interesting words are grouped into topics and how the DBN is constructed.

In its latest version, the ConceptNet project scans online documents such as Wikipedia and compiles a common sense network that links concepts with oriented relations. For example, the concepts Duck and Bird are in the relation IsA (Figure 6). In order to use the stemmed words produced by the interest metric with ConceptNet, first we use the Natural Language Toolkit (Loper, Bird 2002) to convert the word stems into lemmas. The reference vocabulary for retrieving the lemmas is created from the original transcriptions. The subsystem that groups words by topic uses the vocabulary formed by the words of interest; we merge all vocabularies from all discussions, resulting in a general topic model for the story as it was discussed by all readers. This model can be then applied to particular discussions for tracking topics. It is important to note that this general vocabulary of interesting words is much smaller than the vocabulary of the transcriptions, allowing us to use Bayesian networks, since the number of nodes is relatively small.

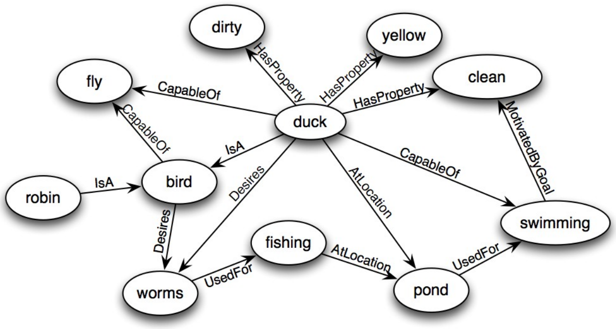


Figure 6. A small example of nodes and relations in ConceptNet.

We define a topic as a set of words relating to a common theme, without any particular order. Topics do not have names themselves and are described only by the words that belong to them. This allows us to model each topic through the common sense relations between its constituent words.

The algorithm that constructs the topics starts by removing a word from the vocabulary of interest and creating a new topic containing only that word. Then it keeps removing one word at a time and, using ConceptNet, calculating the distance between that word and all current topics. More specifically, using the Divisi toolkit from ConceptNet we compute the distance between the current word and each word in the topic using the singular value decomposition matrix (Havasi, Speer, Alonso 2009). If the value is above a threshold (i.e. the distance is small enough), the result counts as a positive vote that the word should belong to the topic. If a high enough percentage of words already in the topic give a positive vote, the new word is included in the topic. The process repeats itself until there are no more words in the vocabulary of interest. We give examples of topics in the results section.

Tracking topics with Dynamic Bayesian Networks

In this section we discuss how we track topics using Dynamic Bayesian Networks (DBN). DBNs are a type of temporal Bayesian networks. For example, a Hidden Markov Model (HMM) is a particular case of a DBN. As with HMMs, a DBN has a sensor, or observed layer, and a latent layer. A DBN has two types of probability models: a sensor model and a transition model. The sensor model describes the probability of the latent nodes, given the current observation evidence nodes. The transition model gives the probability for the values of the latent nodes knowing their values at the previous step, in the absence of any information about the sensor nodes. These two models are combined in a ratio to produce the latent values for the current time step (Murphy 2002).

We model the topics as latent nodes and their constituent words as sensor nodes, with all nodes having binary values. (Figure 7).

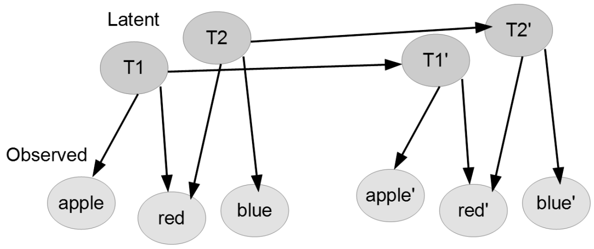


Figure 7. Sample DBN showing the network structure.

For a given time step, a sensor node is true if set of active words from the interest metric file contains the node's corresponding word at that time step. If it doesn't, the node counts as being observed as false. At every time step, all sensor nodes are observed as either true or false. The arrows from the latents to the sensor nodes correspond to the topics. Since the topics are latent and we can not make any other assumptions about their interaction, we assume that latents don't influence other latents between time steps.

Using the Bayes Network Toolkit for Matlab, we construct a DBN using the topic groupings and train it with the expectation maximization algorithm using the interest metric files. We have only one topic grouping that corresponds to all the gathered data and we consider each interest metric file sequence as a separate series of evidence. Once this model is created, it can be used to track topics across particular sessions (Stage 3 in Figure 1).

This topic model allows us to have a general understanding of what a discussion topic is while being able to identify it in a discussion in which only a subset of all its constituent words are present. This will allow us to make more refined suggestions for expanding discussions and to have a better understanding of the intent behind the speakers' words.

Results

In this section we will present two sets of results: first on crowdsourced annotation performance and second on grouping words by topic in conjunction with different parameter settings.

Annotation Results

We used a number of strategies for annotation. The transcription accuracy is given as a percentage of how closely the Mechanical Turk results matched a professional transcription.

The baseline approach was to use the first transcription directly (i.e. results from the first HIT). This approach results in one iteration per task and an average accuracy of 60%. For our purposes, the resulting transcriptions are not very useful.

Next, we used the output of an automated speech recognition system, as a starting suggestion which the workers were asked to improve. We gained no significant improvements overall using this method. However, it is interesting to note that the number of fragments marked as noise decreased when given some text to improve on.

The third approach we used was to iterate until the text would not change significantly between iterations, always giving the last version to improve on. This approach improved accuracy to 80%, at the cost of processing each audio clip. The highest transcription cost was associated with exclamations (“aha”, “yay”, “hum”, etc.), usually reaching the maximum number of iterations, since workers would type them differently every time.

Finally, we noticed that the time of the day at which we submitted the tasks influenced the results. The average accuracy of transcribing the same 92 audio segments is shown in Figure 8 relative to EST. Each batch of 92 files took approximately four hours to annotate, and we hypothesize that the difference observed in accuracy may be partially due to the availability of fluent English speakers during different times of the day.

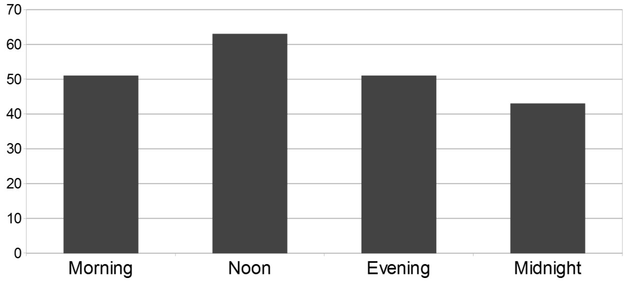


Figure 8. Average accuracy for non iterative transcriptions depending on the time of day that the tasks were submitted.

In summary, our findings support the work of Parent and Eskenazi (2010) showing that iterative refinement leads to the highest quality crowdsourced text. Additionally, we found that in order to achieve results within a relatively short period of time it is important to maintain a large number of HITs, or else compensate for a low number of HITs through higher pay.

Topic Modeling Results

We evaluated our topic modeling algorithm using two data sets: 1) Transcribed data collected from the story, and 2) Wikipedia biographical articles. In both cases, we had two parameters: how similar two words should be in order to consider them potentially in the same topic, ranging from -1 to 1. This values is taken from the singular value decomposition matrix generated by ConceptNet; the second was the percentage of words from a topic that need to match (are similar enough) in order to include the word in their topic.

For the data collected with the story, using 0.5 minimum similarity and 50% vote, we found the following topics (minimum size of 2 words per topic):

- ant owl bird ladybug duck
- noise tap
- color blue purple green yellow
- need say let want
- ladybug bird beetle
- happen let pass
- cricket bird
- purple different green
- yummy hungry
- firefly ladybug
- push angry

The results consist of many small topics, with some words mixed between topics, for example “ant” and “ladybug” being grouped with birds. Increasing the voting percentage to 85% and lowering the minimum similarity to 0.3, we found the following topics using the same data:

- owl bird duck
- push tap
- blue purple different green yellow
- ant firefly cricket ladybug beetle
- push say let pass
- need let want
- push happen let pass
- yummy hungry

This set of parameters resulted in an improved semantic grouping, particularly for nouns. We observed this trend repeat itself, that topics that make more sense to humans are produced by having many roughly similar words in one topic instead of having few highly similar along with other less similar words per topic. Usually very high similarities exist only between words each having few meanings, which is not the case for this application.

Our second set of tests utilized biographical articles from Wikipedia. We used the same interest metric for extracting relevant words in sequence and obtained the following topic groupings.

For JFK's early life biography[1]:

- toilet hospital health
- week old ten make late year
- school grade graduate harvard college student
- member boy father brother family
- member ambassador american father boy
- europe city boston ranch connecticut germany london america
- john joe jack
- rise return help move send health become include
- member palm ten
- swim sail football send jack team

However, we observed an interesting effect. Since in biographies names and years are frequent and occur in groups, the same interest metric used for tracking words of interest can be used to extract most dates and important names in sequence. For the above topic grouping we removed these very specific words since they are not present in ConceptNet.

Conclusions

In this paper we present a system aimed at understanding common topics of discussion among parents and children sharing an interactive tablet experience. Extracting this data over numerous reading sessions allows us to create a general model that can identify trends in particular sessions which might have scarce data. We applied this model to conversations between parents and their pre-school children. These interactions blend educational and entertainment purposes in an interactive tablet application that uses short puzzles and games designed for pre-literacy children. In this collaborative exploration of the application, parents have a guiding role, helping their children. In future work, we will develop methods for merging topics across multiple families. By modeling and understanding their common intentions and preferences for discussion, we hope that our system can enhance the discussions that parents have with their children.

More generally, as highlighted by our use of Wikipedia articles, the presented approach is entirely domain independent. We therefore believe that in future work the techniques presented here could be applied to analysis of either spoken or written text in a broad range of applications, including gaming and other entertainment media. For example, topic modeling can be used not only to provide new topic suggestions, but can also be used to cluster similar discussions or create user profiles.

[1] http://en.wikipedia.org/wiki/John_F._Kennedy

References

Chang, A., Breazeal, C., 2011. TinkRBook: Shared Reading Interfaces for Storytelling. In *The 10th International Conference on Interaction Design and Children*, Ann Arbor, MI

Eisenstein, J., Barzilay, R., 2008. Bayesian unsupervised topic segmentation. In *EMNLP '08 Proceedings of the Conference on Empirical Methods in Natural Language Processing:* 334-343. Stroundsburg, PA.

Hausendorf, H., Quasthoff, U., 1992. Patterns of adult-child interaction as a machanism of discourse acquisition. *Journal of Pragmatics 17:* 241-259.

Havasi, C., Speer, R., Alonso, J., 2009. ConceptNet: A lexical resource for common sense knowledge. In Recent Advances in Natural Language Processing, Volume 5, John Bernjamins Publishers.

Krause, A., Leskovec, J., Guestrin, C., 2006. Data association for topic intensity tracking. In *ICML '06 Proceedings of the 23rd international conference on Machine Learning*, 497-504: New York, NY.

Lamere, P, Kwok, P., Gouvea, E., Raj, B., Singh, R, Walker, W., Wolf, P., 2003, The CMU Sphinx-4 Speech Recognition System, Carnegie Mellon University, Pittsburgh, PA.

Levenshtein, Vl, 1965. "Binary codes capable of correcting deletions, insertions, and reversals". *Soviet Physics Doklady* 10: 707–10.

Loper, E., Bird, S., 2002. NLTK: the Natural Language Toolkit. In *ETMTNLP '02 Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics – Volume 1*, 63-70, Stroudsburg, PA.

Murphy, K. P. 2002. Dynamic Bayesian Networks, Ph.D. diss, Computer Science, University of California, Berkley, CA

Parent, G., Eskenazi, M., 2010. Toward better crowdsourced transcription: Transcription of a year of the Let's Go Bus Information System data. In *Proceedings of Spoken Language Technology Workshop (SLT)*:312-317. Pittsburgh, PA.

Porter, M., 2011. Snowball: A language for stemming algorithm. Online at http://www.snowball.tartarus.org

Reckman, H., Orkin, J., and Roy, D., 2011. Extracting aspects of determiner meaning from dialogue in a virtual world environment. *Proceedings of the International Conference on Computational Semantics (IWCS)*. Oxford, England.

Reckman, H., Orkin, J., and Roy, D., 2010. Learning meanings of words and constructions, grounded in a virtual game. *Proceedings of the 10th Conference on Natural Language Processing (KONVENS)*. Saarbrücken, Germany.