Modeling Discussion Topics in Interactions with a Tablet Reading Primer

Adrian Boteanu
Worcester Polytechnic Institute
100 Institute Road Worcester, MA 01609
aboteanu@wpi.edu

Sonia Chernova
Worcester Polytechnic Institute
100 Institute Road Worcester, MA 01609
soniac@wpi.edu

ABSTRACT

CloudPrimer is a tablet-based interactive reading primer that aims to foster early literacy skills and shared parent-child reading through user-targeted discussion topic suggestions. The tablet application records discussions between parents and children as they read a story and leverages this information, in combination with a common sense knowledge base, to develop discussion topic models. The long-term goal of the project is to use such models to provide context-sensitive discussion topic suggestions to parents during the shared reading activity in order to enhance the interactive experience and foster parental engagement in literacy education. In this paper, we present a novel approach for using commonsense reasoning to effectively model topics of discussion in unstructured dialog. We introduce a metric for localizing concepts that the users are interested in at a given moment in the dialog and extract a time sequence of words of interest. We then present algorithms for topic modeling and refinement that leverage semantic knowledge acquired from ConceptNet, a commonsense knowledge base. We evaluate the performance of our algorithms using transcriptions of audio recordings of parent-child pairs interacting with a tablet application, and compare the output of our algorithms to human-generated topics. Our results show that words of interest and discussion topics selected by our algorithm closely match those identified by human readers.

Author Keywords

Context-Aware Computing, User and Cognitive Models, Dialog Analysis

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User-centered design

General Terms

Design, Human Factors

IUI'13, March 19–22, 2013, Santa Monica, CA, USA.

INTRODUCTION

Currently there is significant interest in applications which track user interest and preferences while adapting to better suit individual needs. Such systems are found in numerous fields, ranging from commercial applications such as online shopping to research in adaptive tutoring ([5], [21]). CloudPrimer is a tablet-based interactive media and reading primer that seeks to foster early literacy skills by enhancing shared parent-child reading. The application is designed to provide an interactive reading experience and to guide readers through various game-like tasks, such as mixing colors and counting. The target age of the children is 3 to 5 years old. 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. We have developed an Android-based reading primer that audio records the conversation that takes place between the parent and child as they read an interactive story book. The long-term goal of our work is to facilitate greater parent-child engagement by learning discussion patterns from numerous interactions and to use these patterns during a live reading for suggesting new topics of conversation to broaden the parent-child interaction.

Modeling dialog topics in informal discussion presents a particular set of challenges. In free form discussions, the goals are not agreed upon in advance. Since the participants do not announce detailed intentions on how they expect the interaction to evolve, abrupt changes of subject are frequent. In this relaxed form, expanding the discussion in a direction from which all participants can benefit from the most takes precedence over debating a well defined topic. The main drivers of the conversation are the participants' interests, their relationship and any external inputs, such as a book or movie that they might be discussing.

The context and development of such interactions contrast with formal settings, such as written articles and news broadcasts. The latter have a distinct, professional, approach in handling a subject and describing it. Speakers in this context try to meet the expectations of a large public who does not offer real time feedback, so stating opinions and facts accurately is important. This is accomplished through a crisp discourse which uses names and jargon to anchor the readers' or viewers' focus of attention. In contrast, a casual conversation uses improvised references that are mostly relevant only for the other participants [11].

We designed a system for segmenting, annotating and analyzing spoken dialog data in order to model discussion topics that occur naturally in informal conversation. Taking into account the particular characteristics of such conversations, we incorporate common sense reasoning in evaluating the discussions.

Our system is divided into three major processing components: 1) gathering and annotating the dialog data, 2) modeling the discussion topic, and 3) making real time discussion topic suggestions to enhance the interactions. Our dialog corpus consists of parent-child reading discussions recorded at a local preschool, which have been transcribed to text using Amazon's Mechanical Turk [14]. This paper focuses on the second processing component, modeling the topic of discussion, which will form the basis for the topic suggestion system in our future work. The purpose of the topic modeling component is to summarize the discussion by tracking the readers' interests and produce topics that would seem reasonable to the average person if he/she were be listening to the conversation.

The main contributions of this paper are:

1. a method of assessing users' interest and focus as it changes over the course of the dialog

2. a two-phase topic modeling system that leverages commonsense reasoning

Both contributions are context independent and do not rely on background information about the dialog.

EDUCATIONAL CONTEXT

Early literacy is a term used to describe the stage of literacy development that occurs before children are able to read and write. During this stage, important abilities include cognitive skills, vocabulary development, phonemic awareness and letter knowledge. The development of early literacy skills through early experiences with books and stories has been critically linked to a childs reading and academic success. Repeated studies have shown that reading aloud to children and providing opportunities for them to discuss the stories that they hear is of utmost importance [2].

In recent years, electronic books have been introduced, aiming to promote early literacy and increase child engagement by including animation and sound effects in the stories. Scientific evaluations of these technologies have found that, although engaging, such devices do not effectively achieve educational goals when used alone [8]. Instead, recent studies highlight the importance of joint parent-child reading, showing that learning gains are achieved by combining the use of digital media with adult interaction [18].

Our system aims to foster early literacy skills through the development of an interactive tablet-based reading primer that promotes parental engagement in reading through the use of a targeted discussion topic suggestion system. The primer supports parents in their role as storytellers and teachers by providing suggestions for open ended discussion; Figure 1 presents a possible example.

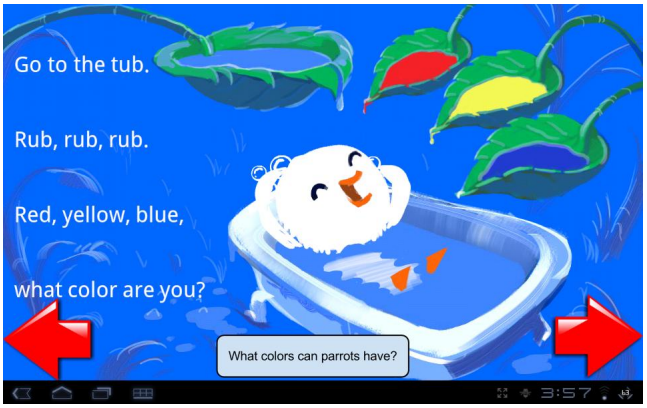

Figure 1. Simulated in-story discussion suggestion via a dialog box.

Generating diverse and relevant discussion topics is critical to the success of this approach. As a result, instead of precoding a set of fixed topics, our work seeks to leverage a community of readers to crowdsource discussion topic suggestions using recorded interactions to suggest topics for new readers. Specifically, the Cloud Primer application 1) records discussions of parent-child pairs engaged in a reading activity, 2) identifies topics of discussion, 3) builds statistical models of discussion topic transitions, and 4) provides appropriately timed suggestions of discussion topics based on data gathered from across the community of readers. This paper focuses on the second challenge of identifying topics of discussion in transcribed speech.

Modeling topics requires understanding the particular interests of the readers as individuals and as a group. Compared to more traditional printed forms, interactive stories are rich in visual media and contain few words, both printed and spoken. Thus gathering enough personalized context information requires transcriptions of the verbal parent-child interactions. Otherwise the potential discussion topic suggestions would be limited by the relatively rigid and simple story structure.

Performing topic suggestions in real time poses many challenges. Speech recognition is highly unreliable for small children, thus our project relies on crowdsourcing through Amazon Mechanical Turk for speech transcription. Recent work [10] shows that audio transcription by non-experts on Mechanical Turk can be achieved with latency as low as 4 seconds with accuracy over 90%, which is sufficiently close to real time for our application. We anticipate that topic suggestions will lag approximately 10 seconds behind the discussion in the final application due to processing delays. Effective interface design, timing of suggestions and wording of suggestions pose additional future challenges for this work.

RELATED WORK

Several intelligent systems which learn from users have been designed in numerous areas. Such solutions include automated office managers [13], personified assistants [17] and smart house applications [1]. These systems either interact directly with the user, through messages or an avatar, or can

change their behavior without explicitly notifying the user. The common approach between all these numerous applications is to create a personalized model by observing an individual user. Our work focuses on extracting relevant patterns from a large population. This will allow our system to have a more general and robust representation of what is relevant for a new reader.

Significant work has been done in tracking topics in news, scientific literature and meetings by using data mining methods and machine learning [9, 4]. 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.

In this paper we introduce a metric for identifying and tracking words that that are relevant to the topic of the conversation. Existing metrics for finding relevant words, such as TF-IDF [16], rely on finding words that have a high frequency in a small subset of documents, differentiating them from other documents. Our project's final goal is to find both common discussion topics and their variations and make suggestions of how to enhance a given interaction based on previous experience.

The bag of words model is a common approach for simplifying text [20]. While the interest metric presented in this paper also reduces the transcriptions to significant sets of words, the key difference is that our approach produces a localized time sequence, as opposed to a single set of words that summarizes the entire text. Furthermore, the topic tracking component of our system takes into account the order in which words occur. Our algorithm for grouping words into topics also has some similarity with K-Means approaches for clustering words [19]. Both our raw topic formation and topic refinement techniques can be viewed as two clustering steps in a nonlinear space, using two different metrics (SVD distance and path length). However, the key differences are the absence of a cluster center and having the same word belong to multiple clusters. Because of these factors, our approach of topic formation is not affected by the initial starting words.

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 toolkit which supports many practical textual-reasoning tasks over real-world documents, including topic-gisting, analogy-making, and other context oriented inferences [7]. ConceptNet forms a large graph of concepts connected through relations. From the ConceptNet project, we use the Divisi2 toolkit, which is a sparse singular value decomposition matrix of the graph relations, to measure the pairwise similarity distance between words; we also use the graph data directly.

Finally, in selecting the significant words, for this project we use a method loosely inspired on human short term memory for identifying words of interest that will be grouped into topics. Since the crowdsourced transcriptions already contain significant noise, with the prospect of speech recognition introducing even more noise if used, implementing a complex cognitive system such as ACT-R [3] would be ineffective. Thus, our method implements a simpler function that does not attempt to mimic human cognition.

OVERVIEW OF SYSTEM ARCHITECTURE

In this section, we present a high level description of our system; Sections Identifying Readers' Interest, Raw Topic Formation and Topic Refinement present the core contributions of this work in greater detail. Figure 2 presents the sequence of modules and the data flow. In Table 1 we present a running example that illustrates how the data is processed by each component.

- Dialog Data Collection: represents the crowdsourcing system used for data collection and annotation. Its output is a line by line transcription of the dialogue. This transcription may contain noise, such as misheard words or typos. A short example of this transcription is presented on the first row of Table 1.

- Preprocessing: prepares the transcriptions to be evaluated by the interest metric module. We preprocess the text by removing all stop words and other very common English words. Then we use the Snowball stemming engine [15] on all words. After this stage, the normalized text mostly consists of nouns, verbs and adjectives, all in stem form. Row two of Table 1 contains the result of preprocessing the dialog sample.

- Interest Metric Function: reads the preprocessed text in sequence, one word at a time, and outputs a set of words that, according to our algorithm, the readers are interested in. We consider each word from the preprocessed text a different time step. At each new time step, all words of interest are output as set. An example of how the output set changes through four consecutive moments in time is shown on row three of Table 1. Through this set of words we represent the current concepts that the speakers are interested in as they are talking.

- Topic Vocabulary Creation: creates a set of all words of interest selected across the entire interaction based on the output of the interest metric module above. We convert the word stems to lemmas using the vocabulary built at Stage 2 and build a vocabulary of lemmas, which can be searched in ConceptNet.

- Raw Topic Formation: constructs the raw topics based on the vocabulary of lemmas and the Divisi2 module of ConceptNet. The output of this module presents a starting point for refining topics.

- Topic Refinement: refines topics by exploring ConceptNet's graph structure. Unlike to the previous step, this process involves directly traveling across edges to identify the connected sub-graphs.

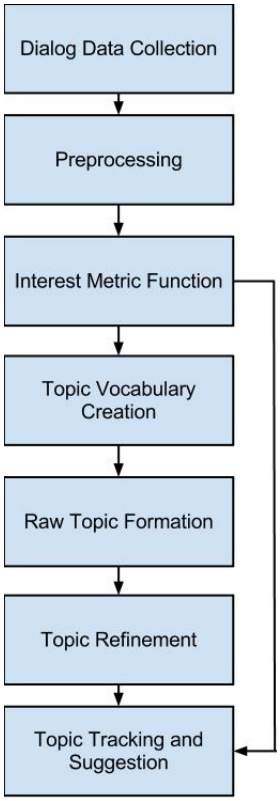

Figure 2. System block diagram.

- Topic tracking and suggestion: This module is a work in progress and is not described in detail in this paper. Using the above generated topics, we construct a Dynamic Bayesian Network in which each word from the vocabulary of lemmas built at Stage 4 is represented as a sensor node. The sensor nodes are connected to latent nodes, each representing a topics, according to the topics produced by Stage 6. We then apply EM on this network using the metric files resulted from Stage 3 to learn the network's parameters. Then, using the generated network, we plan to track topics across other discussions and make recommendations based on the inferred current topic.

IDENTIFYING READERS' INTEREST

The first step in grouping words by topic is to identify which words are the most relevant to the discussion topic. 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 [6]. We observed that parents often have to restate the goals and important concepts to keep their children on track. Another characteristic of dialog is that the density of topic-relevant words, such as nouns, verbs and adjectives, is low compared to a written document. The sequence in which the words are spoken is important for our purposes, since the system must both track topics and make suggestions for developing the discussion as it progresses during the interaction. Finally, crowdsourcing the audio data annotation with Amazon Mechanical Turk introduces noise due to misheard words, worker fatigue or laziness, providing an additional challenge.

Algorithm Stage	**Data**
1. Dialog Data Collection	Here you go, you have a purple duck. What's your favorite color? Um, purple. So baby duck is hungry, eat two beetles. The beetles? Yeah, that's the beetles. More, more. Those are ladybugs.
2. Preprocessing	purpl duck favorit color um purpl babi duck hungri eat beetl beetl yeah beetl ladybug want beetl need beetl need beetl want feed
3. Interest Metric Function)	Time step 0: {purple tap duck} Time step 1: {purple tap duck beetle} Time step 2: {purple beetle tap duck yeah} Time step 3: {purple beetle tap duck}
4. Topic Vocabulary Creation	{purple tap duck beetle yeah}
5. Raw Topic Formation	{blue purple green yellow different} {ant firefly cricket ladybug beetle} {owl bird duck} {need let want}
6. Topic Refinement	{blue purple green yellow} {different} {ant firefly cricket ladybug beetle} {owl bird duck} {need want} {let}

Table 1. Example of the results after each processing stage.

In this section, we introduce a metric for measuring how interested the user is in words occurring in dialog. Our scope is different from document categorization techniques that process a vast collection of documents searching for words that are both relevant and suitable for discerning among the collection. Instead, we track common words based on how significant they might be to the particular users when reading the story. One premise of the metric is that more frequent words, that have a higher local density, have greater importance. Particular for our scenario, parents often repeat words more than it would be necessary in an adult discussion, so that their children learn or focus on them [6]. Our second assumption is that if a word was spoken frequently during the interaction, it will regain a similar level importance in the conversation if it is mentioned again after a longer period of time. This assumption is inspired from the concept of short term memory, although our system does not attempt to replicate the cognitive process.

Interest Metric Function

We use a set structure to capture all unique word stems and associate each with two values, the interest metric value, m, and 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 3); new words are added to the set with a default initial metric value. The frequency of a word, f, is calculated in *updateFrequency*, as the number of the word occurrences divided by the total number of words in the interval between the first and the most recent occurrence of the word.

The interest metric value is updated by iterating through all the words in the set and calling the update function (Figure 4)

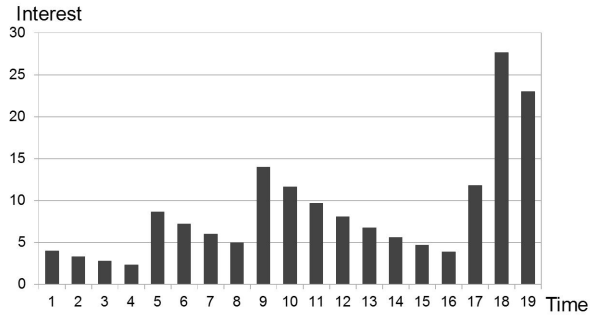

Figure 3. Example on how the interest metric value changes for a word as it is encountered at steps 1, 5, 9, 17 and 18. Note that for steps 17 and 18 the metric value is compensated for very frequent past occurrences.

on each word. If the current word does not match the word in the set, the interest metric value is lowered. If a match is found, the first step is to check (by comparing with the two thresholds on line 2) if the word was frequently spoken before but the current metric value is low. The value of f at this point reflects the previously calculated local frequency of the word. If these conditions are met, the interest metric value is compensated to take into account the readers talking about a word, although the word has not been spoken recently. Requiring both the metric value to be low and the previous local frequency to be high prevents the compensation being applied to common words that have a relatively high frequency but an uniform distribution.

On line 4 the metric function is then increased to mark the word's occurrence. This increase is applied to any word, if it occurs at the present time step. We then compute the new relative frequency over the entire span, from the first occurrence. This limits the number of times a word's interest metric is compensated because, as the discussion progresses, the frequency will gradually approach the average frequency of the word in the text. This enables elements that are present throughout the story, such as the protagonist, to be gradually replaced by newer and more local concepts, making it possible to offer the readers more varied suggestions that follow the story better. For the same purpose, we limit the maximum interest metric value a word can have.

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

Figure 4. The algorithm for calculating the interest metric value. Lines 2 and 3 simulate the recent memories that the speakers would have about past discussions. For all the results presented here we used $a = 2, b = 2, c = 2, d = 1.1$ and $e = 0$, after taking into account the average number of words per sentence after preprocessing.

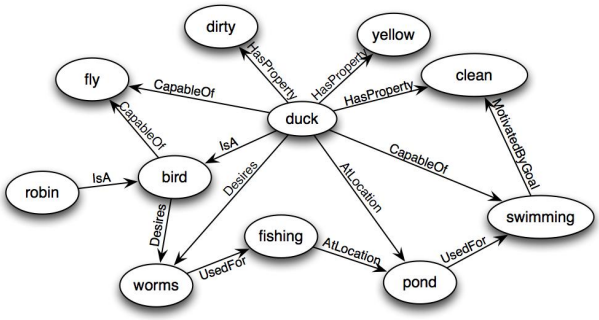

Figure 5. A small example of nodes and relations in ConceptNet

Topic Vocabulary Creation

After reading a new word and updating the interest metric values of the entire set, the module outputs the set of words that have the interest metric value higher than a given threshold. We consider these words to be relevant, or interesting, to the reader at that particular moment. The result is a sequence of sets of words, one per time step (example in row 3 of Table 1). At a given step, the output set can be void if no word has a high enough metric value.

In the final processing stage, words are converted from stem form to lemmas using the spell checking vocabulary with the NLTK library [12]. We also construct the vocabulary of words of interest using the entire metric file. This vocabulary forms the starting point for creating topics, which we present in sections Raw Topic Formation and Topic Refinement.

RAW TOPIC FORMATION

In this section, we describe the first stage of grouping the vocabulary of interest into discussion topics. This step produces rougher topics that are later refined. An example of a set of raw topics is given in Table 1.

The main source of knowledge for our topic creation algorithm is ConceptNet. 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, in the small concept network shown in Figure 5, the concepts 'Duck' and 'Bird' are in the relation 'IsA'. In order to use the stemmed words produced by the interest metric with ConceptNet, first we use the Natural Language Toolkit [12] 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 (20-30). 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 defined only by the words that belong to them. This allows us to model

each topic through the common sense relations between its constituent words. One key characteristic is that a word can belong to multiple topics at once. We allow this since words usually have multiple meanings.

The algorithm that constructs topics starts by removing a word from the vocabulary of interest and creating a new topic containing only that word. For all the remaining words in the set, we remove each word one at a time and, using ConceptNet, calculate the similarity distance between that word and all current topics. More specifically, using the sparse matrix Divisi2 toolkit from ConceptNet we compute the distance between the current word and each word in the topic using the singular value decomposition matrix [7]. 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 can control the specificity of each topic by adjusting both the minimum similarity threshold as well as the minimum percentage of votes. Figure 6 shows the base algorithm. Divisi2 returns a similarity value between two concepts, in the interval between -1 (completely opposite) and 1(identical). For example, by using a high minimum similarity (0.5) but a

$topics = \phi$ //the set of topics to be created
$v = readVocabulary()$ //the vocabulary of interesting words
$for\ each\ w\ in\ v$:
  $for\ each\ t\ in\ topics$: //t is a topic:
    $similar = 0$ //number of votes from words from t
    //tw is a word belonging to the topic t
    $for\ tw\ in\ t$:
      $if\ (similarity(w, tw) > similarityThreshold)$:
        $similar = similar + 1$
    $if\ similar > size(t) * minSimilarityVote$:
      $t = t \cup w$
  $if\ w\ was\ not\ added\ to\ any\ existing\ topic$:
    $topics = topics \cup w$ //create a new topic from w

Figure 6. Pseudocode for raw topic creation.

low voting percentage (50%) loose topics are generated, such as the following.

- 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

Using a lower similarity threshold (0.3) but a higher minimum voting percentage (85%) produces tighter topics, mostly because a newly introduced word has to have some association with most other words present in the topics, such as the following example:

- 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

TOPIC REFINEMENT

The method we introduce in the previous section produces imprecise results. However, it has the advantage of speed over directly exploring the highly connected ConceptNet graph, segmenting the initially very large search space into smaller regions. In this section we introduce a method of refining those results by directly following the graph structure of ConceptNet, which traverses relation edges between concept nodes. ConceptNet contains two types of relations:

- Set relations – some examples are shown in Figure 5: 'IsA', 'AtLocation', etc.
- Concept relations, which have any form – for example, concepts 'Bread' and 'Sandwich' are linked by relation 'be in'.

As with any exploration of very large search spaces, the computational complexity increases dramatically with the depth of the search. From our experience, nodes with a degree of 30 or higher are common. The key idea of refining topics is to find connected components of the subgraph represented by the raw topic in ConceptNet.

We consider two concepts to belong to the same topic after refinement if there is a path between the two respective connected components of at most the length of the search depth. For example, if concepts 'Blue' and 'Red' are both connected by the 'Is A' relation to the concept 'Color', a search with the depth of 2 will consider them belonging to the same topic. Figure 7 shows in pseudocode the algorithm we propose for directly exploring the ConceptNet graph for refining topics.

The advantage of this approach is that it can eliminate any spurious associations introduced by approximating relations with Divisi2, without the high cost of searching in the initial large vocabulary. In our evaluation we found that the most practical search depth is 2. Table 2 shows how the search

$p = \phi$
$for\ w\ in\ t:$
  candidates$\leftarrow$*the set of all nodes connected to w by a path of length d or less*
  $split \leftarrow True$
  for q in p: //q is a refined topic:
    $if\ q \cap candidates = \phi:$
      $q \leftarrow q \cup \{w\}$
      $split \leftarrow False$
  $if split = True:$
    $p = p \cup \{w\}$

Figure 7. Pseudocode for refining topics. t is the topic that is being refined, p is the resulting set of topics after separating the words from t and d is the search depth.

Raw topic	{**deer wing frog duck**}	{**owl bird duck**}	{**push happen let pass**}
Depth = 1	{deer}{wing} {frog}{duck}	{owl}{bird} {duck}	{push}{pass} {happen}{let}
Depth = 2	{deer}{wing} {frog duck}	{owl}{bird duck}	{push}{pass} {happen}{let}
Depth = 3	{deer}{wing} {frog duck}	{owl}{bird duck}	{push}{pass} {happen let}
Depth = 4	{deer} {wing duck} {frog duck}	{owl bird duck}	{push} {happen let pass}

Table 2. Topic refinement results for increasing search depths.

depth affects refining a small topic. In all these examples all types of relations are taken into account.

We found that restricting the search to only some relation types (for example traversing only relation edges 'IsA', 'CapableOf' and 'UsedFor') does not produce better results. Doing so leads to unpredictable output for a given depth because of missing or noisy links. In particular, concept relations, since they do not form a closed set, are difficult to account for in advance through preset filters. For example, one might expect concepts 'Owl' and 'Duck' to be both linked to 'Bird' by the 'Is A' relation, but as Table 2 shows, it takes more than 2 steps to associate them.

RESULTS EVALUATION

To evaluate the performance of our algorithms, we compared the results from three surveys conducted with human participants. In each of the surveys, the participants were presented with either a transcription or a set of words and were asked to answer questions regarding the topic of conversation or word grouping.

Interest metric

To evaluate the performance of our interest metric function, we compared its ability to pick out relevant words in a transcribed conversation to that of human readers. Participants were given short transcriptions of a conversation and were instructed to list five to ten words that captured the main topics of discussion. No information about the broader story plot was provided. In Table 3 we compare the output vocabulary of the interest metric function both with the union and the intersection of the survey responses; 14 participants provided responses to the first transcription, and 10 for the second.

Row 1 from Table 1 presents a short excerpt from one of the transcriptions used. The second column of Table 3 (Interest metric function) shows the vocabulary of important words produced by the interest metric algorithm when applied to the entire transcription. The third column contains the union of all words used by participants. Words that appear in both columns two and three are marked in bold. The fourth column presents the percentage of words from the metric function results that are in bold. Last two columns of the table show the words that were common between all survey respondents and the match percentage with our results.

Comparing our results with the union of all words used by the respondents shows the agreement between our results and the survey responses. The reduced size of the intersection of all survey vocabularies is an indicator of the overall disagreement between respondents. Since there is little agreement between the respondents and the word set is too small to reflect story details, there is no single commonly accepted answer to which we can compare our algorithm's results. However, trying to fully match the vocabulary union of all answers would lead to the inclusion of irrelevant words. Thus, we consider that our results provide a good compromise of general and specific words.

As shown in the 'Match union' column, a significant percentage of words identified by our interest metric function were also present in the survey answers. We found that part of the discrepancy between the columns can be attributed to the fact that (despite specific instructions not to do so) some survey participant answers included words that were not in the text, such as 'number', 'animal' and 'teach'. While these words accurately summarize parts of the discussion, they do not appear in the transcription (e.g., the parent is teaching the child counting and colors, but the word "teach" does not occur in the conversation). Unlike the human participants, our algorithm can only select words of interest from the transcribed words, which is why the density of bold words is higher in the second column compared to the third.

A second observation that we can make is that, while our interest metric module does not discriminate between nouns, verbs and adjectives, human respondents preferred nouns and adjectives to verbs and names. 57% of the words selected by the interest metric that were not matched by participant responses were verbs and names. Among words matched by participants, only 17% were verbs or names. This trend may suggest that humans focus on nouns the most when trying to get the gist of a conversation, though for our particular educational purpose we believe it is important to track all spoken words.

Overall, our interest metric output successfully matches a significant proportion from the words identified by human readers, while avoiding unrepresentative words.

#	Interest metric function	Survey answers vocabulary union	Match union	Survey answers intersection	Match intersection
1	**duck tap color beetle yellow walk ant** say want pass need happen **blue** different **yummy** anymore **purple angry** hurray **bird firefly ladybug cricket** let **baby eat play hungry green push grass**	**angry** animal **ant baby** back **beetle bird blue** butterfly centipede child **color cricket** day done dripping **duck eat** egg finger **firefly** fly frog game go good **grass green** have here help **hungry ladybug** learn **let** lullaby mouth music night open owl page parenthood past **play purple push** read red screen software **tap** teach touch **walk** words **yellow yummy**	70%	**bird tap duck purple**	100%
2	help **brown color** fox **puddle deer** slap **four baby rim** eat **cookie duck** run **arm** even use **feed** let buddy buggy floor six please three next rub **splash** call lot wet white tell **swim blue play good** hit may dive plug little gonna rain job **eight yellow** chris **tub** hold **day care count** kind name **bug many** work **hungry** know **page** turn nine fun five **want wing** talk jack	**feed swim color puddle duck deer** bath **four good** number learn children **day want** toys **count blue hungry arm baby bug splash** teach **white care** teen **brown play** ball giver book story parent you water **cookie tub** red **yellow rim wing** sound animals **eat many** dog **eight** tudy teach learn children school care **page** clap	42%	**duck cookie color**	100%

Table 3. Comparison between keywords identified by our approach, on the left, and keywords identified by human respondents, when given the same text to analyze.

Topic Refinement

We developed two surveys to evaluate the results of our topic refinement algorithm.

The first survey required participants to look through a list of words and select a subset of words that forms a common topic. For example, when presented with the set of words {*brown old long hello thing green yellow okay yes whole white red* } a possible response would be {*brown green yellow white red*}. In total 12 such topics were evaluated through surveys and the results were compared with the topic refinement algorithm output for search depths of 2 and 3. There were 12 participants in this survey, each evaluating only two topics from the 12 possible, amounting to 24 evaluations in total. Having a few evaluations per participant reduced the bias that a particular participant might introduce in the results. The answers for a question were considered to be in agreement only if they were identical.

The average agreement between the survey participants was 66%, which indicates a high probability of respondents disagreeing on selecting words to refine a topic. Table 4 shows, on average, how the survey results compare to our system's output. The table reports the percentage of participants for whom the topic was a *subset* of, exactly the *same* as, a *superset* of, or mostly *different* from the algorithm output. Table 5 shows sample results results of our system's output after topic refinement to participant responses and the classification result used in computing Table 4; for a given topic, the outputs for both search depths are included. These results show that a search dept of 3 produces topics with low cohesion since 58% of the survey results are subsets of the refined topic.

Search Depth	Subset	Same	Superset	Different
2	8.34%	41.66%	25%	25%
3	58.3%	0%	16.7%	25%

Table 4. Average topic refinement results compared survey responses.

The second survey consisted of a set of "odd word out" problems in which respondents were given a short list of words and instructed to select the one that did not fit with the others. The option 'None' was also available in the case that the participants considered that all words were similar. In total there were 26 topics. On average, each topic received 12 different evaluations from participants.

The average agreement between participants was 73.5%. The survey answers can be divided into two groups by agreement, high and low. For the nine topics which contained mostly verbs, the agreement ranged between 30% to 60% with an average of 45.4%. For the rest of the 17 topics, mostly formed by nouns, the agreement ranged between 75% and 100%, with a mean of 88.4%. This high agreement group of topics contains words that are interpreted similarly by the reviewers. In contrast, topics with low agreement contain words with multiple meanings, thus subjective to evaluate. Using a search depth of 2, the topic refinement algorithm matched the dominant decision of the survey answers for 47% of the topics – it either eliminated the same word or kept the topic unchanged. For a search depth of 3, the percentage is 29%. Table 6 presents a few examples of the survey questions, show-

Raw Topic	Survey Response	Depth 2 Result	Depth 2 Class	Search Depth 3 Result	Depth 3 Class
brown old long hello thing green yellow okay yes whole white red	brown green yellow white red	brown green yellow white red	same	brown hello thing green yellow okay yes whole white red	subset
chew tuck stop put let lay touch follow hello	chew tuck lay touch	(no topic)	different	chew lay follow hello	different
seven ten six three next four thing cant nine five eight hello	seven ten six three four nine five eight	three four nine five eight	superset	three next four nine five eight hello	subset

Table 5. Sample results between our approach for refining raw topics (second column) with survey responses.

Raw Topic	Outliers - Topic Refinement	Outliers - Survey Responses
Blue Purple Different Green Yellow	Depth 2, 3: Different	**Different (13/15)** Purple(1/15) None(1/15)
Ant Firefly Cricket Ladybug Beetle	Depth 2: Cricket Depth 3: None	Ant(2/15) Firefly(1/15) Cricket(2/15) Beetle(1/15) **None (9/15)**
Push Say Let Pass	Depth 2, 3: Push Say Let Pass (No common topic found)	Push(1/15) **Say(8/15)** Let(1/15) None(2/15)
Need Want Let	Depth 2, 3: Let	Need(1/15) Let(7/15) None(7/15)
Deer Wing Frog Duck	Depth 2, 3: Deer, Duck	**Wing(12/13)** Duck(1/13)

Table 6. Comparison between our approach and survey responses for refining raw topics. For the survey responses, the number of agreeing responses is shown as a fraction of the total responses for that particular question. If there is one, the predominant decision is in bold.

ing the raw topics, the words selected as outliers by the topic refinement algorithm, and words selected as outliers by the survey participants. Based on these results, we can conclude that our system best matches human respondents when analyzing topics composed of nouns. An exception to this is the example in the last row of Table 6, in which the algorithm was unable to differentiate animals (deer, frog, duck) from a limb (wing). We attribute such errors to currently missing edges in the constantly expanding commonsense knowledge network.

The most significant disagreement, both between our system and the respondents, and between the respondents themselves, occurs on topics consisting of verbs, such as in lines 3 and 4 of Table 6. These collections are more difficult to interpret, and refining the topic would imply adopting a specific angle. For example, on line 3 of the table, the consensus in selecting 'Say' might be that it is the only action that produces speech, but similar classifications can be found to eliminate other words.

This evaluation shows that, for the situations in which human respondents reach consensus on the constituency of a topic, our approach successfully matches that consensus.

CONCLUSIONS AND FUTURE WORK

The system described and demonstrated in this paper allows us to track readers' interest across a conversation and to build dialog topic models. It is a core component of a larger system designed to enhance conversations through real time topic suggestions. This will supplement the current touch input tablet application with the capability to offer intelligent feedback to it's users, making the learning experience more effective while retaining it's the game-like approach.

The methods we introduce are independent of the content of the discussion. When compared to human respondents' replies to the same task, our system's output is similar for both important word selection and topic generation. Because the survey respondents mostly agree with our system's output, we believe that the later stages of the project will provide meaningful hints for parents to develop conversations with their children.

We present an approach to track individual sessions of interaction with the tablet primer. By unifying all vocabularies of interest for all interactions of a parent with their child, we can build a general topic model for that family. We currently have work in progress to merge topics across families and create a generalized topic model for a story. This topic model, which will reflect meaningful trends of what families find interesting to talk about while reading the story, will be used to track topics and make suggestions in the deployment phase of the project.

Acknowledgments

We thank Cynthia Breazeal, David Nunez and Angela Chang for providing the interactive reading primer. This work was supported by the National Science Foundation award number 1117584.

REFERENCES

1. Bouchard, B., Bouzouane, A., and Giroux, S. A smart home agent for plan recognition. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, AAMAS '06, ACM (New York, NY, USA, 2006), 320–322.
2. Burns, M., Griffin, P., and Snows, C. Starting out right. a guide promoting children's reading success. wdc, 1999.
3. Douglass, S., and Ball, J. *Large Declarative Memories in ACT-R*. 2009, 222–227.
4. Eisenstein, J., and Barzilay, R. Bayesian unsupervised topic segmentation, 2008.
5. Feng, M., Heffernan, N., and Koedinger, K. Addressing the testing challenge with a web-based e-assessment system that tutors as it assesses. In *WWW '06 Proceedings of the 15th international conference on World Wide Web* (2006), 307–316.
6. Hausendorf, H., and Quasthoff, U. Patterns of adult-child interaction as a machanism of discourse acquisition. *Journal of Pragmatics 17* (1992), 241–259.
7. Havasi, C., Speer, R., and Alonso, J. *ConceptNet: A lexical resource for common sense knowledge*, vol. 5. John Benjamins Publishing Company, 2009.
8. Korat, O., and Shamir, A. The educational electronic book as a tool for supporting children?s emergent literacy in low versus middle ses groups. *Computers & Education 50*, 1 (2008), 110–124.
9. Krause, A., and Guestrin, C. Data association for topic intensity tracking. Tech. rep., In International Conference on Machine Learning (ICML, 2006.
10. Lasecki, W., Song, Y., Kautz, H., and Bigham, J. Real-time crowd labeling for deployable activity recognition, 2012.
11. Linell, P. *Approaching Dialogue*, vol. 1. John Benjamins Publishing Company, Philadelphia, PA, 1998.
12. Loper, E., and Bird, S. Nltk : The natural language toolkit. *Processing 1*, July (2002), 1–4.
13. Modi, J., Veloso, M., Smith, F. S., and Oh, J. Cmradar: A personal assistant agent for calendar management, 2005.
14. Paolacci, G., Chandler, J., and Ipeirotis, P. Running experiments on amazon mechanical turk, 2010.
15. Porter, M. Snowball: A language for stemming algorithms.
16. Ramos, J. Using tf-idf to determine word relevance in document queries, 2003.
17. Rich, C., and Sidner, C. L. Collagen: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction 8* (1998), 315–350. 10.1023/A:1008204020038.
18. Segal-Drori, O., Korat, O., Shamir, A., and Klein, P. Reading electronic and printed books with and without adult instruction: effects on emergent reading. *Reading and Writing 23*, 8 (2010), 913–930.
19. Steinbach, M., Karypis, G., and Kumar, V. A comparison of document clustering techniques. In *In KDD Workshop on Text Mining* (2000).
20. Wallach, H. M. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, ACM (New York, NY, USA, 2006), 977–984.
21. Weld, D., Adar, E., Chilton, L., Hoffmann, R., Horvitz, E., Koch, M., Landay, J., Lin, C., and Mausam, M. Personalized online education a crowdsourcing challenge. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence* (2012).