Identifying Speakers in Children's Stories for Speech Synthesis

Jason Y Zhang, Alan W Black and Richard Sproat

Language Technologies Institute
Carnegie Mellon University
azure@cs.cmu.edu, awb@cs.cmu.edu
and AT&T Labs Research
rws@research.att.com

Abstract

Choosing appropriate voices for synthesizing children's stories requires text analysis techniques that can identify which portions of the text should be read by which speakers. Our work presents techniques to take raw text stories and automatically identify the quoted speech, identify the characters within the stories and assign characters to each quote. The resulting marked-up story may then be rendered with a standard speech synthesizer with appropriate voices for the characters.

This paper presents each of the basic stages in identification, and the algorithms, both rule-driven and data-driven, used to achieve this. A variety of story texts are used to test our system. Results are presented with a discussion of the limitations and recommendations on how to improve speaker assignment in further texts.

1. Introduction

Speech synthesis has now reached the stage where high quality human sounding speech can be synthesized for many applications. Unit selection speech synthesis, where appropriate subword units are selected from large database of natural speech, are best when the domain is known [1], but can still produce pleasant sounding speech in the general case, [2]. However there are restrictions in such systems. The high quality speech is directly linked to the quality of the speech database itself. Unlike previous synthesis technologies like diphone and formant synthesis, unit selection does not currently offer variation outside its recorded style.

This work is part of an NSF grant to investigate more varied synthesis. Specifically we are looking at modeling prosodic variation in reading children's stories where a human reader will add different voice qualities, appropriate intonation, etc, to carry the meaning of the story over to the listener.

In order to do this well, we need to do analysis of the text in order to uncover some of the underlying structure. This paper looks at one particular aspect of text analysis for speech synthesis. We examine how to automatically identify spoken text within a story and by identifying the characters in the story, assign each quote to a particular character. The result is a system, we call ESPER, which can take in raw text stories and produce markup identifying who speaks when. The markup can then be rendered in a speech synthesis markup language like SABLE or SSML and transformed into speech.

Although we are dealing with well-written published texts, there is still significant variety on how the information within the story is presented. Hence identifying the speakers within the story is a non-trivial task, even when we choose to ignore clearly-difficult cases, *e.g.* when detailed semantics of the story or external world knowledge is required for the identification.

2. ESPER: architecture

In order to narrate a children's story using a variety of synthesized voices, ESPER steps through a number of stages to identify the speaker for each piece of quoted speech in the story. It is necessary to first identify all the pieces of spoken speech in the story, as well as all the characters in the story who are potential speakers. Then an association must be made between each piece of quoted speech and the appropriate story character who has spoken it. At each processing step, ESPER encapsulates all the acquired speech information in a markup format such as HTML, Sable (an XML-based speech synthesis markup language) [3], and CSML, (Childrens Story Markup Language), a specially-created Markup language for speech information in children's stories.

ESPER is implemented within the Festival Speech Synthesis framework [4]. Although ESPER itself does not speak, it will be a component of the larger storyteller system. Festival also provides much of the infrastructure that detailed text analysis requires: such as punctuation and tokenization, part of speech tagging, utterance representation, and extraction of data for machine learning techniques. In addition, we make use of Festival's XML support.

2.1. Identifying Spoken Speech in a Children's Story

We define quoted speech to be any quote-annotated text segments within the body of a story. ESPER is able to detect quoted speech in a given story and label the extracted speech using CSML. The following is an example of the quoted-speech CSML markup:

```
<QUOTE TYPE="NEW"> 'Come, there's no use
in crying like that!'</QUOTE> said
Alice to herself, rather sharply;
```

It is worth noting that even though the content of these stories are child-oriented, the structure of the stories can nevertheless be quite complex. Hence ESPER also takes into consideration such structures as nested quoted-speech; this can occur when a character in a story is narrating a story of his or her own, with its own set of characters and quoted speech, essentially creating a story within a story. For example:

```
The farmer described it to his wife:
<QUOTE TYPE="NEW"> "The tail-feathers
of the fowl were very short, and it
winked with both its eyes, and said
<QUOTE TYPE="NEW"> "Cluck, cluck."</QUOTE>
What were the thoughts of the fowl as it
said this I cannot tell you..." </QUOTE>
```

We tested the quoted-speech identification module in ESPER on a single story where all quoted speech had been hand-annotated.

The story was selected from a collection of stories not included in ESPER's development corpus.

Table 1: *Quoted-Speech Identification Evaluated on* Little Women*, Chap.1, By L.M. Alcott*

Recall	Precision
100%	94%

The results show that ESPER was able to correctly identify all the hand-annotated spoken speech in the story. However, the comparatively lower precision is attributable to the fact that ESPER does not discriminate between actual spoken speech and quoted labels. Consider the following:

- ... She opened it, and found in it a very small cake, on which the words **'EAT ME'** were beautifully marked in currants. 'Well, I'll eat it,' said Alice...

In this case, the quoted text seems to represent a label. Although it may be reasonable to synthesize such labels using a special voice to make the story more interesting, we must nevertheless distinguish them from actual pieces of speech so as not to mistakenly assign any random character's voice, which would produce confusion for the story listener. In the future, we would like to automatically differentiate such labels from quoted-speech and synthesize them using a pre-defined voice (such as that of the narrator or the main character).

2.2. Identifying Quoted Speech Types

Not every piece of quoted speech is an independent unit. Just as the words in a sentence are connected, there exist connections between adjacent pieces of quoted speech so that they share similar properties such as being spoken by the same speaker. An important feature of ESPER's quote-identification module is the ability to detect whether a piece of quoted speech is a new quote (NEW), where it is most likely spoken by a different speaker from the previous speaker, or a continuation quote (CONT) where it carries on the sentiments from the previous speech and is spoken by the same speaker as that of the previous. A CSML marked-up example of a new and continuation quote is shown below:

```
<QUOTE TYPE="NEW"> 'Come, there's no use
in crying like that!'</QUOTE> said
Alice to herself, rather sharply;
<QUOTE TYPE="CONT"> 'I advise you to
leave off this minute!' </QUOTE>
```

Identifying the connectivity between quoted speech segments can significantly reduce the amount of work performed when we are identifying the speakers for each piece of quoted speech. For instance, since we make the assumption that the quoted speech segments that are continuations (CONT) share the same speaker as their predecessors, then once we identify the speaker for a quoted speech segment, we can apply the speaker information from that segment to all its continuation segments, thereby reducing computational effort and potential error build-up.

2.3. Using a Decision Tree For Quoted-Speech Type Identification

We trained a decision tree (CART) to identify the aforementioned types of quoted speech using local feature information in the story text. The collection of training data consisted of 16 children's stories taken from works by Hans Christian Andersen and Lewis Carroll, with a total of 1198 pieces of quoted speech. In order to ensure that the training data are correctly labeled, we performed a first approximation of quoted speech types over the training data using a naive rule such that if the first word in the quoted speech is not capitalized, then the quote is classified as type "CONT"; otherwise it is classified as type "NEW". The resulting output from this initial pass was then hand-corrected to eliminate any incorrect type assignment resulting from the application of this rule. From this training data, we then extracted a number of features for each piece of quoted speech in order to train the decision tree. These features include:

- the word string preceding the quote
- the word string succeeding the quote
- capitalization of the first word in the quote
- whether the quote starts the paragraph,
- whether the quote is the first quote in the paragraph,
- the end-punctuation of the previous quote within the same paragraph,
- the punctuation of the word preceding the quote
- the length (number of words) of the gap between the last quote and the current one (within the same paragraph).

The performance of the decision tree after training was as follows:

Table 2: *Performance of Decision Tree on Identifying Type of Quoted Speech*

New	Cont.
98.8%	82.6%

From examining the tree it is interesting to notice that the feature which serves as the most reliable predictor of quoted-speech types is the capitalization feature. Even though intuitively, other features, such as the punctuation of the previous token before the quote, might also seem like good predictors of quote types, statistically they were deemed to be less reliable.

2.4. Character Identification in a Children's Story

It is not enough to merely identify the pieces of speech in a story. In order to model each piece of speech using appropriately different voices, it is vital that we also be able to identify the characters in the story, all of whom are potential speakers. This, in a sense, is a named-entity extraction task, since we first need to identify all the proper names in a particular story, and extract from these Named Entities, only those which are names of characters in the story. For this purpose, we have considered using a Named-Entity Extraction System for the task. Here we evaluated the performance of one of the most commonly-used named-entity extraction systems, the BBN IdentiFinder [5], which can scan through a body of text and locate the names of people, places, and other named entities of interest to the user, and output these entities in a markup format. We tested the BBN IdentiFinder on two manually-labeled children's stories selected for their contrasting stylistic differences. The number of characters in each story is within the range of 14-16. Results are shown in Table 3 and 4.

However, it is not sufficient to confine the scope of character identification to only proper names in the story. For instance, the works of Hans Christian Andersen contain a considerable number of characters who are not named, but merely referred to by descriptions; for example, *the peasant's wife*, *the man with the sheep*. In these cases we would need additional linguistic information to make the proper identification.

To this end, we have created a character identification module within ESPER. This module uses pattern matching to extract proper names from the story, similar to the functionality of the BBN IdentiFinder. In addition, it uses the part-of-speech information derived from the probabilistic POS tagger in the Festival speech synthesis system to extract non-proper names, such as definite noun phrases, as potential character names. Similar to the BBN IdentiFinder, our character identification module also encapsulates the extracted entities in the text using a markup language, specifically the CSML language, where each each character is also assigned an ID for reference purposes, as well as a CLASS to represent its type (i.e., the character name could be a proper name or a definite NP, etc.), which is useful in the speaker identification stage. An example CSML:

```
<CHARACTER ID="LITTLE_TUK" CLASS="properName">
Little Tuk</CHARACTER> sprang out of bed
quickly and read over his lesson in the
book... <CHARACTER ID="THE_OLD_WASHERWOMAN"
CLASS="defNP"> The old washerwoman
</CHARACTER> put her head in at the door,
and nodded to him quite kindly..
```

We tested the performance of both the BBN IdentiFinder and ESPER's character identification module and the results are as follows:

Table 3: *Performance of the BBN IdentiFinder and the ESPER Character-Markup Module on* Alice In Wonderland*, Chapter 3.*

	Recall	Precision
BBN	77.8%	73.7%
ESPER	88.9%	53.2%

Table 4: *Performance of the BBN IdentiFinder and the ESPER Character-Markup Module on* Little Tuk

	Recall	Precision
BBN	61.5%	53.3%
ESPER	76.9%	38.5%

From these results we observe that ESPER performed with a higher recall than the BBN IdentiFinder since it is able to identify more characters within the story. But at the same time, it also retrieves more non-relevant entities in the story as character names, and hence it suffers from a lower precision than the BBN IdentiFinder. However, in our case it is more important to have a higher recall (i.e., not to overlook potential character names). If the actual set of story characters is a subset of what we retrieved, we can utilize some discrimination factors or relevance rankings in later steps of processing to filter out the non-relevant entity names from our retrieval set. On the other hand, if an actual character name is not included in our retrieval set, there is no way to recover them in future steps of processing.

2.5. Identifying the Speakers

The final processing stage is to assign each NEW quote to a speaker from the cast of characters identified in the previous step. We want to find out how well this can be done with the minimum amount of deep analysis.

We analyzed stories within our development corpus to devise conditions that seemed adequate for speaker identification. A number of simplifying assumptions were made that turned out to be very reasonable. First we assumed that the speaker is referenced in the same paragraph where the quoted-speech occurs; this is, in fact, true in a high percentage of cases. We also assumed that character names within the quote are not the speaker even though some speakers refer to themselves in the third person.

Through some basic analysis on test stories we came up with the following simple rule: if there is a character name preceding the quoted speech within the same paragraph, then assign it as the speaker of the quoted speech. Otherwise use the named character following the quoted speech. We have additionally augmented this to favor character names which are proper names, using the character CLASS information acquired from the character-identification stage. However, results have shown that this addition only resulted in a minimal improvement in accuracy (approx. +1%). An example of the speaker markup in CSML is as follows:

```
<QUOTE TYPE="NEW" SPEAKER="ALICE"> 'As wet
as ever,'</QUOTE> said <CHARACTER NAME=
"ALICE" CLASS="properName"> Alice
</CHARACTER> in a melancholy tone:
<QUOTE TYPE="CONT"> 'it doesn't seem to dry
me at all.'</QUOTE> <QUOTE TYPE="NEW"
SPEAKER="THE_DODO"> 'In that case,'</QUOTE>
said <CHARACTER ID="THE_DODO" CLASS="defNP">
the Dodo</CHARACTER>solemnly...
```

ESPER's speaker-identification performance is shown below:

Table 5: *Performance of the ESPER Speaker Identification Module on Two Distinctly-Styled Children's Stories*

Story Name	Accuracy
Alice. Chap.3	86.7%
Little Tuk	47.6%

We observe a rather high variance in ESPER's performance between the two test stories. it is evident that Chapter 3 of *Alice* contains a more consistent speech-speaker relationships, so that ESPER is able to achieve a relatively high accuracy in finding the speakers. On the hand, there is very little regularity in the speech-speaker relationships within *Little Tuk*, which results in a degradation in performance. Note that these two stories were selected for their contrasting stylistic differences in order to demonstrate the difficulties within speaker identification.

Somewhat surprisingly, two criteria that we thought would be necessary to achieve this level of accuracy did not appear to be as important: giving higher preference to character names in the proximity of "speaking" words such as *said*, *cried*, etc., and anaphora resolution. Nevertheless these conditions are not negligible and will be investigated further in future work.

Detailed error analysis indicates that there are several factors hindering speaker identification. The most prominent one seems to be the difficulty of adapting to novel text structures within a story. For example, in the *Alice* test story , there is an entire section which was artistically structured to resemble the shape of a mouse's tail, and some of these stylistic conventions were easily misinterpreted as paragraph breaks by our system. Other relatively minor sources of error include the problems of differentiating quoted labels from real quoted speech (as noted earlier), and resolving speech spoken by characters whose distance to the speech exceeds the paragraph scope.

We observe that almost all cases of errors are attributable to the incorrect character being selected as the speaker. This is preferrable to the alternative scenario where we are unable to find any speakers at all for a given piece of quoted speech.

Hence we are confident that by refining the disambiguation algorithm in ESPER, we can make sizable improvements to the speaker-identification task. Furthermore, given more time we would prefer to be able to train this module, but currently we do not have sufficient data, nor as yet an appropriate model within which to train. However, our current work continues to investigate this matter.

2.6. CSML

In our work we have designed an XML markup language so that we can mark information within our original text preserving its layout. Current we have only used the elements `CSML`, `QUOTE` and `CHARACTER`. `QUOTE` elements support `TYPE` and `SPEAKER` attributes. `CHARACTER` attributes are `ID` (whose values are the same as `SPEAKER` in `QUOTES`, and `CLASS` (which describes the syntactic type of the character name). We will expand this support as necessary.

3. Story Data

We have examined a large number of children's stories for use as data. Among them are two major collections of stories, which are used for development and evaluation purposes. The children's stories by Hans Christian Andersen and works by Lewis Carroll (focusing mainly on *Alice in Wonderland*). Works from these two authors are selected because of their stylistic diversity, as well as their contrasting writing styles, which is useful for testing the performance flexibility of ESPER. Note that for our development/testing corpus, we hand-selected stories with an above average number of quoted speech segments.

Table 6: *Statistics For the H.C. Andersen and L. Carroll Story Collections*

Story Collection	*Total # of Stories*	*Avg # of Words*	*Avg # Quoted Speech Segments*
Andersen	130	2898	23
Carroll	3	20447	539

4. Discussion

While we feel we have made good progress on the problems addressed here children's stories are, by their nature both heterogeneous and complex, which means in turn that there is much more work to be done. Some of the problems that ESPER needs to address in the future are:

4.1. Quoted-Speech Identification

Different authors use different conventions for quoted speech. For example, the H.C. Andersen collections uses double quotes as begin-quote and end-quote, and the single quote for quote-within-quote. In contrast, works by Lewis Carroll uses the back-tick operator (‘) as begin-quote, the single quote as end-quote, and uses double-quotes for quote-within-quote.

4.2. Character Identification

Our current character identification over-generates, which is a good point, but we clearly need more categorization to make better speaker selection possible. Separating out animate objects from inanimate objects would help (though in children's stories we cannot exclude inanimate objects from speaking). We also need to be able to resolve aliases for characters, for example *Alice* versus *the girl*.

4.3. Speaker identification

The identity of the speaker might be subtle, with embedded in self-reference:

```
"Tuk! little Tuk!"</QUOTE> said a voice.
It was a very little person who spoke.
He was dressed as a sailor, and looked
small enough to be a middy, but he was
 not one. <QUOTE TYPE="NEW"> "I bring
you many greetings from Corsor ... ,"
</QUOTE> said <CHARACTER ID="CORSOR">
CLASS="properName"> Corsor;</CHARACTER>
```

It is not obvious how to make the connection between *a voice* and *Corsor*. We believe that there will be limits on what can be done automatically, and that the more adult the stories the more complex this may be, but we still feel we can do better than our current version.

Nested spoken speech can also be hard (e.g., stories within stories, or quotes within quotes). In addition to the syntactic issues in identifying such quotations, there are the more complex issues of identifying the (sub)speaker. There is also the issue of how such speech should be rendered; should it be in the quoted speaker's voice, or in the main speaker's voice but rendered in such a way as to sound like the quoted speaker, or at least different from the main speaker?

5. Conclusions

ESPER offers a basic process for marking up raw text from children's stories with quotes, characters, and identification of who spoke the quoted speech. The framework allows further expansion on the basic rules and trained models that we have currently provided.

The resulting markup can then be rendered as speech with (hand specified) appropriate voices for each character through a standard speech synthesis markup language.

Our future work will be in improving the current coverage with more analysis, in addition to expanding the types of data in our markup. In addition to resolving speakers, we would like to identify the properties of the speaker that might allow automatic selection of an appropriate voice to use as well as the style the speech should be delivered in.

6. Acknowledgments

This work was funded in part by NSF grant "ITR: Prosody Generation for Child Oriented Speech Synthesis". The opinions expressed in this paper do not necessarily reflect those of NSF.

7. References

[1] A. Black and K. Lenzo, "Limited domain synthesis," in *ICSLP2000*, Beijing, China., 2000, vol. II, pp. 411–414.

[2] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal, "The AT&T Next-Gen TTS system," in *Joint Meeting of ASA, EAA, and DAGA*, Berlin, Germany, 1999, pp. 18–24.

[3] R. Sproat, A. Hunt, M. Ostendorf, P. Taylor, A. Black, K. Lenzo, and M. Edgington, "SABLE: A standard for TTS markup," in *International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[4] A. Black, P. Taylor, and R. Caley, "The Festival speech synthesis system," http://festvox.org/festival, 1998.

[5] Daniel M. Bikel, Richard L. Schwartz, and Ralph M. Weischedel, "An algorithm that learns what's in a name," *Machine Learning*, vol. 34, no. 1-3, pp. 211–231, 1999.