Generating Fine-Grained Reviews of Songs From Album Reviews

Swati Tata and **Barbara Di Eugenio**
Computer Science Department
University of Illinois, Chicago, IL, USA
{stata2 | bdieugen}@uic.edu

Abstract

Music Recommendation Systems often recommend individual songs, as opposed to entire albums. The challenge is to generate reviews for each song, since only full album reviews are available on-line. We developed a summarizer that combines information extraction and generation techniques to produce summaries of reviews of individual songs. We present an intrinsic evaluation of the extraction components, and of the informativeness of the summaries; and a user study of the impact of the song review summaries on users' decision making processes. Users were able to make quicker and more informed decisions when presented with the summary as compared to the full album review.

1 Introduction

In recent years, the personal music collection of many individuals has significantly grown due to the availability of portable devices like MP3 players and of internet services. Music listeners are now looking for techniques to help them manage their music collections and explore songs they may not even know they have (Clema, 2006). Currently, most of those electronic devices follow a Universal Plug and Play (UPNP) protocol (UPN, 2008), and can be used in a simple network, on which the songs listened to can be monitored. Our interest is in developing a Music Recommendation System (Music RS) for such a network.

Commercial web-sites such as Amazon (www.amazon.com) and Barnes and Nobles (www.bnn.com) have deployed Product Recommendation Systems (Product RS) to help customers choose from large catalogues of products. Most Product RSs include reviews from customers who bought or tried the product. As the number of reviews available for each individual product increases, RSs may overwhelm the user if they make all those reviews available. Additionally, in some reviews only few sentences actually describe the recommended product, hence, the interest in opinion mining and in summarizing those reviews.

A Music RS could be developed along the lines of Product RSs. However, Music RSs recommend individual tracks, not full albums, e.g. see www.itunes.com. Summarizing reviews becomes more complex: available data consists of album reviews, not individual song reviews (www.amazon.com, www.epinions.com). Comments about a given song are fragmented all over an album review. Though some web-sites like www.last.fm allow users to comment on individual songs, the comments are too short (a few words such as "awesome song") to be counted as a full review.

In this paper, after presenting related work and contrasting it to our goals in Section 2, we discuss our prototype Music RS in Section 3. We devote Section 4 to our summarizer, that extracts comments on individual tracks from album reviews and produces a summary of those comments for each individual track recommended to the user. In Section 5, we report two types of evaluation: an intrinsic evaluation of the extraction components, and of the coverage of the summary; an extrinsic evaluation via a between-subject study. We found that users make quicker and more informed decisions when presented with the song review summaries as opposed to the full album review.

2 Related Work

Over the last decade, summarization has become a hot topic for research. Quite a few systems were developed for different tasks, including multi-document summarization (Barzilay and McKeown, 2005; Soubbotin and Soubbotin, 2005; Nastase, 2008).

Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1376–1385,
Uppsala, Sweden, 11-16 July 2010.

What's not to get? Yes, **Maxwell**, and **Octopus** are a bit silly! ...
.....
"Something" and **"Here Comes The Sun"** are two of George's best songs ever (and **"Something"** may be the single greatest love song ever). **"Oh Darling"** is a bluesy masterpiece with Paul screaming.....
.....
"Come Together" contains a great riff, but he ended up getting sued over the lyrics by Chuck Berry.....

Figure 1: A sample review for the album "Abbey Road"

Whereas summarizing customer reviews can be seen as multi-document summarization, an added necessary step is to first extract the most important features customers focus on. Hence, summarizing customer reviews has mostly been studied as a combination of machine learning and NLP techniques (Hu and Liu, 2004; Gamon et al., 2005). For example, (Hu and Liu, 2004) use associative mining techniques to identify features that frequently occur in reviews taken from `www.epinions.com` and `www.amazon.com`. Then, features are paired to the nearest words that express some opinion on that feature. Most work on product reviews focuses on identifying sentences and polarity of opinion terms, not on generating a coherent summary from the extracted features, which is the main goal of our research. Exceptions are (Carenini et al., 2006; Higashinaka et al., 2006), whose focus was on extracting domain specific ontologies in order to structure summarization of customer reviews.

Summarizing reviews on objects different from products, such as restaurants (Nguyen et al., 2007), or movies (Zhuang et al., 2006), has also been tackled, although not as extensively. We are aware of only one piece of work that focuses on music reviews (Downie and Hu, 2006). This study is mainly concerned with identifying descriptive patterns in positive or negative reviews but not on summarizing the reviews.

2.1 Summarizing song reviews is different

As mentioned earlier, using album reviews for song summarization poses new challenges:
a) Comments on features of a song are embedded and fragmented within the album reviews, as shown in Figure 1. It is necessary to correctly map features to songs.
b) Each song needs to be identified each time it is referred to in the review. Titles are often abbreviated, and in different ways, even in the same review – e.g. see *Octopus* for *Octopus's Garden* in Figure 1. Additionally, song titles need not be noun phrases and hence NP extraction algorithms miss many occurrences, as was shown by preliminary experiments we ran.
c) Reviewers focus on both inherent features such as lyrics, genre and instruments, but also on people (artist, lyricist, producer etc.), unlike in product reviews where manufacturer/designer are rarely mentioned. This variety of features makes it harder to generate a coherent summary.

3 SongRecommend: Prototype Music RS

Figure 2 shows the interface of our prototype Music RS. It is a simple interface dictated by our focus on the summarization process (but it was informed by a small pilot study). Moving from window to window and from top to bottom:
a) The top leftmost window shows different devices on which the user listens to songs. These devices are monitored with a UPNP control point. Based on the messages received by the control point, the user activities, including the metadata of the song, are logged.
b) Once the user chooses a certain song on one of the devices (see second window on top), we display more information about the song (third top window); we also identify related songs from the internet, including: other songs from the same album, popular songs of the artist and popular songs of related artists, as obtained from Yahoo Music.
c) The top 25 recommendations are shown in the fourth top window. We use the SimpleKMeans Clustering (Mitchell, 1997) to identify and rank the top twenty-five songs which belong to the same cluster and are closest to the given song. Closeness between two songs in a cluster is measured as the number of attributes (album, artist etc) of the songs that match.
d) When the user clicks on *More Info* for one of the recommended songs, the pop-up, bottom window is displayed, which contains the summary of the reviews for the specific song.

4 Extraction and Summarization

Our summarization framework consists of the five tasks illustrated in Figure 3. The first two tasks pertain to information extraction, the last three to repackaging the information and generating a co-

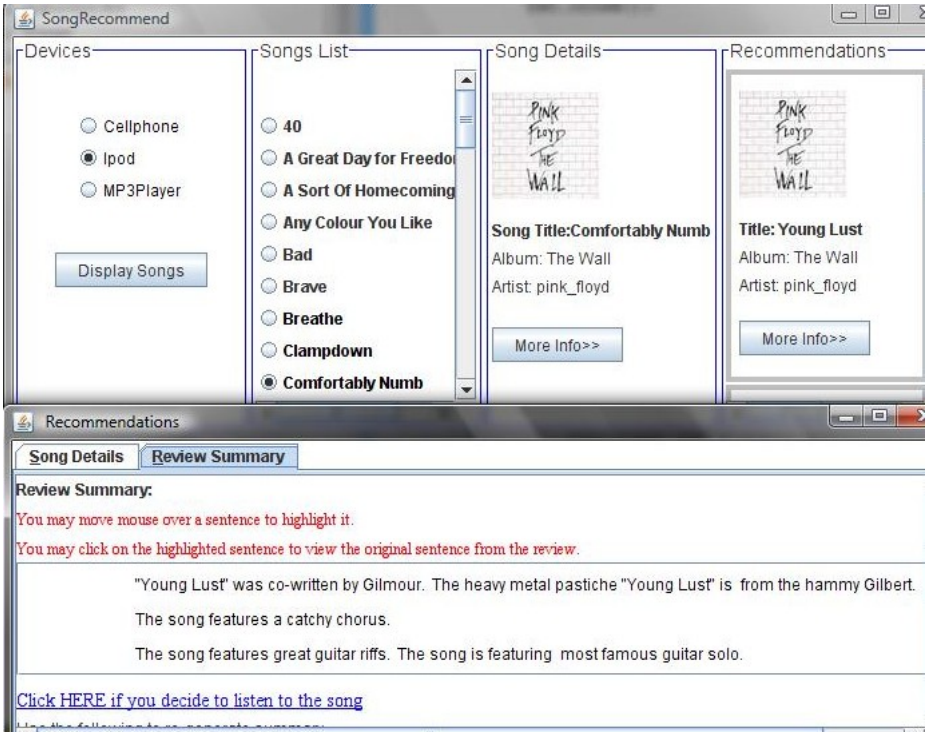

Figure 2: SongRecommend Interface

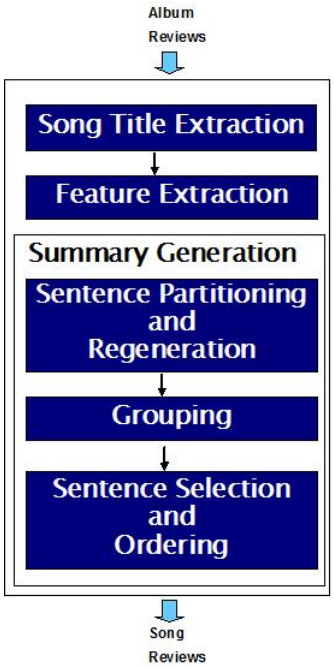

Figure 3: Summarization Pipeline

herent summary. Whereas the techniques we use for each individual step are state-of-the-art, our approach is innovative in that it integrates them into an effective end-to-end system. Its effectiveness is shown by the promising results obtained both via the intrinsic evaluation, and the user study. Our framework can be applied to any domain where reviews of individual components need to be summarized from reviews of collections, such as reviews of different hotels and restaurants in a city.

Our corpus was opportunistically collected from `www.amazon.com` and `www.epinions.com`. It consists of 1350 album reviews across 27 albums (50 reviews per album). 50 randomly chosen reviews were used for development. Reviews have noise, since the writing is informal. We did not clean it, for example we did not correct spelling mistakes. This corpus was annotated for song titles and song features. Feature annotation consists of marking a phrase as a feature and matching it with the song to which the feature is attributed. Note that we have no a priori inventory of features; what counts as features of songs emerged from the annotation, since annotators were asked to annotate for noun phrases which contain "any song related term or terms spoken in the context of a song". Further, they were given about 5 positive and 5 negative

What's not to get? Yes, <**song id=3**>**Maxwell**</**song**>, and <**song id=5**>**Octopus**</**song**> are a bit silly! ...
.....
.....
<**song id=2**>**"Something"**</**song**> and <**song id=7**>**"Here Comes The Sun"**</**song**> are two of <**feature id=(2,7)**>**George's**</**feature**> best songs ever (and <**song id=2**>**"Something"**</**song**> may be
.....
<**song id=4**>**"Oh Darling"**</**song**> is a <**feature id=4**>**bluesy masterpiece**</**feature**> with <**feature id=4**>**Paul**</**feature**> screaming.....
.....
<**song id=1**>**"Come Together"**</**song**> contains a great <**feature id=1**>**riff**</**feature**>, but ...

Figure 4: A sample annotated review

examples of features. Figure 4 shows annotations for the excerpt in Figure 1. For example in Figure 4, *George, Paul, bluesy masterpiece* and *riff* have been marked as features. Ten randomly chosen reviews were doubly annotated for song titles and features. The Kappa co-efficient of agreement on both was excellent (**0.9**), hence the rest of the corpus was annotated by one annotator only. The two annotators were considered to be in agreement on a feature if they marked the same head of phrase **and** attributed it to the same song.

We will now turn to describing the component tasks. The algorithms are described in full in (Tata, 2010).

4.1 Title Extraction

Song identification is the first step towards summarization of reviews. We identify a string of words as the title of a song to be extracted from an album review if it (1) includes some or all the words in the title of a track of that album, and (2) this string occurs in the right context. Constraint (2) is necessary because the string of words corresponding to the title may appear in the lyrics of the song or anywhere else in the review. The string *Maxwell's Silver Hammer* counts as a title only in sentence (a) below; the second sentence is a verse in the lyrics:

a. Then, the wild and weird "Maxwell's Silver Hammer."

b. Bang, Bang, maxwell's silver hammer cam down on her head.

Similar to Named Entity Recognition (Schedl et al., 2007), our approach to song title extraction is based on *n-grams*. We proceed album by album. Given the reviews for an album and the list of songs in that album, first, we build a lexicon of all the words in the song titles. We also segment the reviews into sentences via sentence boundary detection. All 1,2,3,4-grams for each sentence (the upper-bound 4 was determined experimentally) in the review are generated. First, n-grams that contain at least one word with an edit distance greater than one from a word in the lexicon are filtered out. Second, if higher and lower order n-grams overlap at the same position in the same sentence, lower order n-grams are filtered out. Third, the n-grams are merged if they occur sequentially in a sentence. Fourth, the n-grams are further filtered to include only those where (i) the n-gram is within quotation marks; and/or (ii) the first character of each word in the n-gram is upper case. This filters n-grams such as those shown in sentence (b) above. All the n-grams remaining at this point are potential song titles. Finally, for each n-gram, we retrieve the set of IDs for each of its words and intersect those sets. This intersection always resulted in one single song ID, since song titles in each album differ by at least one content word. Recall that the algorithm is run on reviews for each album separately.

4.2 Feature Extraction

Once the song titles are identified in the album review, sentences with song titles are used as anchors to (1) identify *segments* of texts that talk about a specific song, and then (2) extract the feature(s) that the pertinent text segment discusses.

The first step roughly corresponds to identifying the flow of topics in a review. The second step corresponds to identifying the properties of each song. Both steps would greatly benefit from reference resolution, but current algorithms still have a low accuracy. We devised an approach that combines text tiling (Hearst, 1994) and domain heuristics. The text tiling algorithm divides the text into coherent discourse units, to describe the sub-topic structure of the given text. We found the relatively coarse segments the text tiling algorithm provides sufficient to identify different topics.

An album review is first divided into segments using the text tiling algorithm. Let $[seg_1, seg_2, ..., seg_k]$ be the segments obtained. The segments that contain potential features of a song are identified using the following heuristics:

Step 1: Include seg_i if it contains a song title.

These segments are more likely to contain features of songs as they are composed of the sentences surrounding the song title.

Step 2: Include seg_{i+1} if seg_i is included and seg_{i+1} contains one or more *feature terms*.

Since we have no a priori inventory of features (the feature annotation will be used for evaluation, not for development), we use WordNet (Fellbaum, 1998) to identify *feature terms*: i.e., those nouns whose synonyms, direct hypernym or direct hyponym, or the definitions of any of those, contain the terms "music" or "song", or any form of these words like "musical", "songs" etc, for at least one sense of the noun. Feature terms exclude the words "music", "song", the artist/band/album name as they are likely to occur across album reviews. All feature terms in the final set of segments selected by the heuristics are taken to be features of the song described by that segment.

4.3 Sentence Partitioning and Regeneration

After extracting the sentences containing the features, the next step is to divide the sentences into two or more "sub-sentences", if necessary. For example, *"McCartney's bouncy bass-line is especially wonderful, and George comes in with an excellent, minimal guitar solo."* discusses both features *bass* and *guitar*. Only a portion of the sentence describes the *guitar*. This sentence can thus be divided into two individual sentences. Removing parts of sentences that describe another feature, will have no effect on the summary as a whole as the portions that are removed will be present in the group of sentences that describe the other feature.

To derive n sentences, each concerning a single feature f, from the original sentence that covered n features, we need to:

1. Identify portions of sentences relevant to each feature f (partitioning)
2. Regenerate each portion as an independent sentence, which we call f-sentence.

To identify portions of the sentence relevant to the single feature f, we use the Stanford Typed Dependency Parser (Klein and Manning, 2002; de Marnee and Manning, 2008). Typed Dependencies describe grammatical relationships between pairs of words in a sentence. Starting from the feature term f in question, we collect all the nouns, adjectives and verbs that are directly related to it in the sentence. These nouns, adjectives and verbs become the components of the new f-sentence. Next, we need to adjust their number and forms. This is a natural language generation task, specifically, sentence realization.

1. **"Maxwell"** is a bit silly.
2. **"Octopus"** is a bit silly.
3. **"Something"** is George's best song.
4. **"Here Comes The Sun"** is George's best song.
5. **"Something"** may be the single greatest love song.
6. **"Oh! Darling"** is a bluesy masterpiece.
7. **"Come Together"** contains a great riff.

Figure 5: f-sentences corresponding to Figure 1

We use YAG (McRoy et al., 2003), a template based sentence realizer. *clause* is the main template used to generate a sentence. Slots in a template can in turn be templates. The grammatical relationships obtained from the Typed Dependency Parser such as *subject* and *object* identify the slots and the template the slots follows; the words in the relationship fill the slot. We use a morphological tool (Minnen et al., 2000) to obtain the base form from the original verb or noun, so that YAG can generate grammatical sentences. Figure 5 shows the regenerated review from Figure 1.

YAG regenerates as many f-sentences from the original sentence, as many features were contained in it. By the end of this step, for each feature f of a certain song s_i, we have generated a set of f-sentences. This set also contains every original sentence that only covered the single feature f.

4.4 Grouping

f-sentences are further grouped, by sub-feature and by polarity. As concerns sub-feature grouping, consider the following f-sentences for the feature *guitar*:

a. *George comes in with an excellent, minimal guitar solo.*

b. *McCartney laid down the guitar lead for this track.*

c. *Identical lead guitar provide the rhythmic basis for this song.*

The first sentence talks about the *guitar solo*, the second and the third about the *lead guitar*. This step will create two subgroups, with sentence *a* in one group and sentences *b* and *c* in another. We

Let $[f_x\text{-}s_1, f_x\text{-}s_2, ...f_x\text{-}s_n]$ be the set of sentences for feature f_x and song S_y

Step 1: *Find the longest common n-gram (LCN) between f_x-s_i and f_x-s_j for all $i \neq j$: LCN(f_x-s_i, f_x-s_j)*

Step 2: *If LCN(f_x-s_i, f_x-s_j) contains the feature term and is not the feature term alone, f_x-s_i and f_x-s_j are in the same group.*

Step 3: *For any f_x-s_i, if LCN(f_x-s_i, f_x-s_j) for all i and j, is the feature term, then f_x-s_i belongs to the default group for the feature.*

Figure 6: Grouping sentences by sub-features

Example: *The lyrics are the best*
Adjectives in the sentence: *best*

Senti-wordnet Scores of *best*:
Sense 1 (frequency=2):
positive = 0.625, negative =0 , objective = 0.375

Sense 2 (frequency=1):
positive = 0.75, negative = 0, objective = 0.25

Polarity Scores Calculation:
positive(best) = $\frac{2*0.625+1*0.75}{(2+1)}$ = 0.67
negative(best) = $\frac{2*0+1*0}{(2+1)}$ = 0
objective(best) = $\frac{2*0.375+1*0.25}{(2+1)}$ = 0.33

Since the sentence contains only the adjective *best*, its polarity is positive, from:
Max (positive(best), negative(best), objective(best))

Figure 7: Polarity Calculation

identify subgroups via common n-grams between f-sentences, and make sure that only n-grams that are related to feature f are identified at this stage, as detailed in Figure 6. When the procedure described in Figure 6 is applied to the three sentences above, it identifies *guitar* as the longest pertinent LCN between *a* and *b*, and between *a* and *c*; and *guitar lead* between *b* and *c* (we do not take into account linear order within n-grams, hence *guitar lead* and *lead guitar* are considered identical). Step 2 in Figure 6 will group *b* and *c* together since *guitar lead* properly contains the feature term *guitar*. In Step 3, sentence *a* is sentence f_x-s_i such that its LCN with all other sentences (*b* and *c*) contains only the feature term; hence, sentence *a* is left on its own. Note that Steps 2 and 3 ensure that, among all the possible LNCs between pair of sentences, we only consider the ones containing the feature in question.

As concerns polarity grouping, different reviews may express different opinions regarding a particular feature. To generate a coherent summary that mentions conflicting opinions, we need to subdivide f-sentences according to polarity.

We use SentiWordNet (Esuli and Sebastiani, 2006), an extension of WordNet where each sense of a word is augmented with the probability of that sense being positive, negative or neutral. The overall sentence score is based on the scores of the adjectives contained in the sentence.

Since there are a number of senses for each word, an adjective a_i in a sentence is scored as the normalized weighted scores of each sense of the adjective. For each a_i, we compute three scores, positive, as shown in Formula 1, negative and objective, which are computed analogously:

$$pos(a_i) = \frac{freq_1 * pos_1 + ... + freq_n * pos_n}{(freq_1 + ... + freq_n)} \quad (1)$$

a_i is the i^{th} adjective, $freq_j$ is the frequency of the j^{th} sense of a_i as given by Wordnet, and pos_j is the positive score of the j^{th} sense of a_i, as given by SentiWordnet. Figure 7 shows an example of calculating the polarity of a sentence.

For an f-sentence, three scores will be computed, as the sum of the corresponding scores (positive, negative, objective) of all the adjectives in the sentence. The polarity of the sentence is determined by the maximum of these three scores.

4.5 Selection and Ordering

Finally, the generation of a coherent summary involves selection of the sentences to be included, and ordering them in a coherent fashion. This step has in input groups of f-sentences, where each group pertains to the feature f, one of its subfeatures, and one polarity type (positive, negative, objective). We need to select one sentence from each subgroup to make sure that all essential concepts are included in the summary. Note that if there are contrasting opinions on one feature or subfeatures, one sentence per polarity will be extracted, resulting in potentially inconsistent opinions on that feature to be included in the review (we did not observe this happening frequently, and even if it did, it did not appear to confuse our users).

Recall that at this point, most f-sentences have been regenerated from portions of original sen-

tences (see Section 4.3). Each f-sentence in a subgroup is assigned a score which is equivalent to the number of features in the original sentence from which the f-sentence was obtained. The sentence which has the lowest score in each subgroup is chosen as the representative for that subgroup. If multiple sentences have the lowest score, one sentence is selected randomly. Our assumption is that among the original sentences, a sentence that talks about one feature only is likely to express a stronger opinion about that feature than a sentence in which other features are present.

We order the sentences by exploiting a music ontology (Giasson and Raimond, 2007). We have extended this ontology to include few additional concepts that correspond to features identified in our corpus. Also, we extended each of the classes by adding the domain to which it belongs. We identified a total of 20 different domains for all the features. For example, *[saxophone,drums]* belongs to the domain *Instrument*, and *[tone, vocals]* belong to the domain *Sound*. We also identified the priority order in which each of these domains should appear in the final summary. The ordering of the domains is such that first we present the general features of the song (e.g. *Song*) domain, then present more specific domains (e.g. *Sound, Instrument*). $f-$sentences of a single domain form one paragraph in the final summary. However, features domains that are considered as sub-domains of another domain are included in the same paragraph, but are ordered next to the features of the parent domain. The complete list of domains is described in (Tata, 2010). f-sentences are grouped and ordered according to the domain of the features. Figure 8 shows a sample summary when the extracted sentences are ordered via this method.

> "The Song That Jane Likes" is cute. The song has some nice riffs by Leroi Moore. "The Song That Jane Likes" is also amazing funk number.
>
> The lyrics are sweet and loving.
>
> The song carries a light-hearted tone. It has a catchy tune. The song features some nice accents.
>
> "The Song That Jane Likes" is beautiful song with great rhythm. The funky beat will surely make a move.
>
> It is a heavily acoustic guitar-based song.

Figure 8: Sample summary

5 Evaluation

In this section we report three evaluations, two intrinsic and one extrinsic: evaluation of the song title and feature extraction steps; evaluation of the informativeness of summaries; and a user study to judge how summaries affect decision making.

5.1 Song Title and Feature Extraction

The song title extraction and feature extraction algorithms (Sections 4.1 and 4.2) were manually evaluated on 100 reviews randomly taken from the corpus (2 or 3 from each album). This relatively small number is due to the need to conduct the evaluation manually. The 100 reviews contained 1304 occurrences of song titles and 898 occurrences of song features, as previously annotated.

1294 occurrences of song titles were correctly identified; additionally, 123 spurious occurrences were also identified. This results in a precision of 91.3%, and recall of 98%. The 10 occurrences that were not identified contained either abbreviations like *Dr.* for *Doctor* or spelling mistakes (recall that we don't clean up mistakes).

Of the 898 occurrences of song features, 853 were correctly identified by our feature extraction algorithm, with an additional 41 spurious occurrences. This results in a precision of 95.4% and a recall of 94.9%. Note that a feature (NP) is considered as correctly identified, if its head noun is annotated in a review for the song with correct ID.

As a baseline comparison, we implemented the feature extraction algorithm from (Hu and Liu, 2004). We compared their algorithm to ours on 10 randomly chosen reviews from our corpus, for a total of about 500 sentences. Its accuracy (40.8% precision, and 64.5% recall) is much lower than ours, and than their original results on product reviews (72% precision, and 80% recall).

5.2 Informativeness of the summaries

To evaluate the information captured in the summary, we randomly selected 5 or 6 songs from 10 albums, and generated the corresponding 52 summaries, one per song – this corresponds to a test set of about 500 album reviews (each album has about 50 reviews). Most summary evaluation schemes, for example the Pyramid method (Harnly et al., 2005), make use of reference summaries written by humans. We approximate those gold-standard reference summaries with 2 or 3 critic reviews per album taken from `www.pitchfork.`

com, www.rollingstone.com and www.allmusic.com.

First, we manually annotated both critic reviews and the automatically generated summaries for song titles and song features. 302, i.e., 91.2% of the features identified in the critic reviews are also identified in the summaries (recall that a feature is considered as identified, if the head-noun of the NP is identified by both the critic review and the summary, and attributed to the same song). 64 additional features were identified, for a recall of 82%. It is not surprising that additional features may appear in the summaries: even if only one of the 50 album reviews talks about that feature, it is included in the summary. Potentially, a threshold on frequency of feature mention could increase recall, but we found out that even a threshold of two significantly affects precision.

In a second evaluation, we used our Feature Extraction algorithm to extract features from the critic reviews, for each song whose summary needs to be evaluated. This is an indirect evaluation of that algorithm, in that it shows it is not affected by somewhat different data, since the critic reviews are more formally written. 375, or 95% of the features identified in the critic reviews are also identified in the summaries. 55 additional features were additionally identified, for a recall of 87.5%. These values are comparable, even if slightly higher, to the precision and recall of the manual annotation described above.

5.3 Between-Subject User Study

Our intrinsic evaluation gives satisfactory results. However, we believe the ultimate measure of such a summarization algorithm is an end-to-end evaluation to ascertain whether it affects user behavior, and how. We conducted a between-subject user study, where users were presented with two different versions of our Music RS. For each of the recommended songs, the baseline version provides *only* whole album reviews, the experimental version provides the automatically generated song feature summary, as shown in Figure 2. The interface for the baseline version is similar, but the summary in the bottom window is replaced by the corresponding album review. The presented review is the one among the 50 reviews for that album whose length is closest to the average length of album reviews in the corpus (478 words).

Each user was presented with 5 songs in succession, with 3 recommendations each (only the top 3 recommendations were presented among the available 25, see Section 3). Users were asked to select at least one recommendation for each song, namely, to click on the url where they can listen to the song. They were also asked to base their selection on the information provided by the interface. The first song was a test song for users to get acquainted with the system. We collected comprehensive timed logs of the user actions, including clicks, when windows are open and closed, etc. After using the system, users were administered a brief questionnaire which included questions on a 5-point Likert Scale. 18 users interacted with the baseline version and 21 users with the experimental version (five additional subjects were run but their log data was not properly saved). All users were students at our University, and most of them, graduate students (no differences were found due to gender, previous knowledge of music, or education level).

Our main measure is time on task, the total time taken to select the recommendations from song 2 to song 5 – this excludes the time spent listening to the songs. A t-test showed that users in the experimental version take less time to make their decision when compared to baseline subjects ($p = 0.019$, $t = 2.510$). This is a positive result, because decreasing time to selection is important, given that music collections can include millions of songs. However, time-on-task basically represents the time it takes users to peruse the review or summary, and the number of words in the summaries is significantly lower than the number of words in the reviews ($p < 0.001$, $t = 16.517$).

Hence, we also analyzed the influence of summaries on decision making, to see if they have any effects beyond cutting down on the number of words to read. Our assumption is that the default choice is to choose the first recommendation. Users in the baseline condition picked the first recommendation as often as the other two recommendations combined; users in the experimental condition picked the second and third recommendations more often than the first, and the difference between the two conditions is significant ($\chi^2 = 8.74$, $df = 1$, $p = 0.003$). If we examine behavior song by song, this holds true especially for song 3 ($\chi^2 = 12.3$, $df = 1$, $p < 0.001$) and song 4 ($\chi^2 = 5.08$, $df = 1$, $p = 0.024$). We speculate that users in the experimental condition

are more discriminatory in their choices, because important features of the recommended songs are evident in the summaries, but are buried in the album reviews. For example, for Song 3, only one of the 20 sentences in the album review is about the first recommended song, and is not very positive. Negative opinions are much more evident in the review summaries.

The questionnaires included three common questions between the two conditions. The experimental subjects gave a more positive assessment of the length of the summary than the baseline subjects ($p = 0.003$, $t = -3.248$, $df = 31.928$). There were no significant differences on the other two questions, feeling overwhelmed by the information provided; and whether the review/summary helped them to quickly make their selection.

A multiple Linear Regression with, as predictors, the number of words the user read before making the selection and the questions, and time on task as dependent variable, revealed only one, not surprising, correlation: the number of words the user read correlates with time on task ($R^2 = 0.277$, $\beta = 0.509$, $p = 0.004$).

Users in the experimental version were also asked to rate the grammaticality and coherence of the summary. The average rating was 3.33 for grammaticality, and 3.14 for coherence. Whereas these numbers in isolation are not too telling, they are at least suggestive that users did not find these summaries badly written. We found no significant correlations between grammaticality and coherence of summaries, and time on task.

6 Discussion and Conclusions

Most summarization research on customer reviews focuses on obtaining features of the products, but not much work has been done on presenting them as a coherent summary. In this paper, we described a system that uses information extraction and summarization techniques in order to generate summaries of individual songs from multiple album reviews. Whereas the techniques we have used are state-of-the-art, the contribution of our work is integrating them in an effective end-to-end system. We first evaluated it intrinsically as concerns information extraction, and the informativeness of the summaries. Perhaps more importantly, we also ran an extrinsic evaluation in the context of our prototype Music RS. Users made quicker decisions and their choice of recommendations was more varied when presented with song review summaries than with album reviews. Our framework can be applied to any domain where reviews of individual components need to be summarized from reviews of collections, such as travel reviews that cover many cities in a country, or different restaurants in a city.

References

Regina Barzilay and Kathleen McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Giuseppe Carenini, Raymond Ng, and Adam Pauls. 2006. Multi-document summarization of evaluative text. In *Proceedings of EACL*.

Oscar Clema. 2006. *Interaction Design for Recommender Systems*. Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, July.

Marie-Catherine de Marnee and Christopher D. Manning. 2008. Stanford Typed Dependencies Manual. http://nlp.stanford.edu/software/dependencies_manual.pdf.

J. Stephen Downie and Xiao Hu. 2006. Review mining for music digital libraries: Phase ii. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 196–197, Chapel Hill, NC, USA.

Andrea Esuli and Fabrizio Sebastiani. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC-06, the 5th Conference on Language Resources and Evaluation*, Genova, IT.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.

Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. 2005. Pulse: Mining customer opinions from free text. In *Advances in Intelligent Data Analysis VI*, volume 3646/2005 of *Lecture Notes in Computer Science*, pages 121–132. Springer Berlin / Heidelberg.

Frederick Giasson and Yves Raimond. 2007. Music ontology specification. Working draft, February. http://pingthesemanticweb.com/ontology/mo/.

Aaron Harnly, Ani Nenkova, Rebecca Passonneau, and Owen Rambow. 2005. Automation of summary evaluation by the Pyramid method. In *Proceedings of the Conference on Recent Advances in Natural Language Processing*.

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Meeting of the Association for Computational Linguistics*, Las Cruces, NM, June.

Ryuichiro Higashinaka, Rashmi Prasad, and Marilyn Walker. 2006. Learning to Generate Naturalistic Utterances Using Reviews in Spoken Dialogue Systems. In *COLING-ACL06*, Sidney, Australia.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of KDD*, Seattle, Washington, USA, August.

Dan Klein and Christopher D. Manning. 2002. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems 15*, pages 3–10.

Susan McRoy, Songsak Ukul, and Syed Ali. 2003. An augmented template-based approach to text realization. In *Natural Language Engineering*, pages 381–420. Cambridge Press.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust, applied morphological generation. In *Proceedings of the 1st International Natural Language Generation Conference*.

Tom Mitchell. 1997. *Machine Learning*. McGraw Hill.

Vivi Nastase. 2008. Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Patrick Nguyen, Milind Mahajan, and Geoffrey Zweig. 2007. Summarization of multiple user reviews in the restaurant domain. Technical Report MSR-TR-2007-126, Microsoft, September.

Markus Schedl, Gerhard Widmer, Tim Pohle, and Klaus Seyerlehner. 2007. Web-based detection of music band members and line-up. In *Proceedings of the Australian Computer Society*.

M. Soubbotin and S. Soubbotin. 2005. Trade-Off Between Factors Influencing Quality of the Summary. In *Document Understanding Workshop (DUC)*, Vancouver, BC, Canada.

Swati Tata. 2010. *SongRecommend: a Music Recommendation System with Fine-Grained Song Reviews*. Ph.D. thesis, University of Illinois, Chicago, IL.

2008. UPnP Device Architecture Version 1.0. (www.upnp.org).

Li Zhuang, Feng Jing, and Xiaoyan Zhu. 2006. Movie review mining and summarization. In *Conference on Information and Knowledge Management*, Arlington, Virginia, USA.