Natural Language Engineering **20** (1): 29–67. © Cambridge University Press 2012
doi:10.1017/S1351324912000253

SongRecommend: From summarization to recommendation

SWATI TATA[1*] and BARBARA DI EUGENIO[2]
[1]*Versay Solutions, Chicago, IL, USA*
e-mail: swtata@gmail.com
[2]*Department of Computer Science, University of Illinois, Chicago, IL, USA*
e-mail: bdieugen@uic.edu

(*Received 21 July 2011; revised 24 August 2012; accepted 28 August 2012; first published online 28 September 2012*)

Abstract

In recent years, the availability of too much information has become a fact of life for anybody connected with the Internet. The same is true for music: because of the penetration of portable devices and the availability of millions of tracks on the web, individual music collections have become unwieldy. Users need tools to help search their own song collections, and to recommend songs they may be interested in. Whereas recommendation systems have been developed for a variety of products, a music recommendation system presents special challenges, including the ability to recommend individual songs, as opposed to entire albums, even if only full album reviews are available on-line. SongRecommend, our music recommendation system, combines information extraction and generation techniques to produce summaries of reviews of individual songs from album reviews. We present a number of evaluations for SongRecommend: intrinsic evaluations 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. When presented with the summary, users were able to make quicker decisions, and their choices were more varied. Whereas the smaller size of the summary has an impact on time-on-task, users do not appear to choose a specific recommendation only based on number of words. Our work demonstrates that state-of-the-art techniques in Natural Language Processing can be integrated into an effective end-to-end system.

1 Introduction

In just over sixty years, music fruition has undergone at least three revolutions. In 1948, the vinyl LP was successfully introduced by Columbia, at the same time that magnetic tapes, precursors to audiocassettes, were being perfected. In 1982, the audio CD and CD players became available. The latest revolution is due to the penetration of portable and non-portable devices like iPods, MP3 players, SmartPhones, of which

* This research was conducted while Dr. Tata was a PhD student at the University of Illinois, Chicago, and was initially supported by Motorola, Inc. Dr. Di Eugenio was partially supported by award IIS-0905593 from the National Science Foundation. Thanks to the anonymous reviewers for constructive comments and suggestions.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

most users own more than one; and to the availability of millions of tracks on the web. These two concauses have dramatically increased the size of the personal music collections of many individuals. As noted in Celma (2006, 2010), users are no longer able to *effectively* search their own large digital collections, or peruse what is digitally available on the Internet. Recommender systems can benefit users by assisting them in filtering their collections and discovering music they may not even know they have. Music popularity follows distribution with a *Long Tail* (Celma 2010; Goel *et al.* 2010; Levy and Bosteels 2010): for example, in 2007 1 percent of all digital tracks accounted for 80 percent of all sales; 1,000 albums accounted for 50 percent of all album sales, and 80 percent of all albums sold were purchased less than 100 times. Not only should an effective recommender system return music that is interesting and relevant, but should also find tracks that are *novel* for the user (Levy and Bosteels 2010).

Naturally, recommender systems are needed well beyond music, since the explosion of information is affecting every possible realm of user interest. Commercial web sites, such as Amazon (www.amazon.com) and Barnes and Nobles (www.bnn.com), have been deploying Product Recommendation Systems (Product RS) for a few years now. To help new customers assess the merits of a product, most Product RSs include reviews from customers who have experience with that product. However, the increasing number of reviews available for each individual product compounds the general problem of overwhelming the user with too much information. In addition, some reviews are long, and only few sentences actually concern the recommended product: hence, the interest in opinion mining and in summarizing those reviews (Hu and Liu 2004; Gamon *et al.* 2005; Carenini, Ng and Pauls 2006; Zhuang *et al.* 2006; Nguyen, Mahajan and Zweig 2007).

It would be feasible to develop a Music RS following the same approaches that were used for Product RSs. However, recommending music is different from recommending cameras, restaurants, and also books or movies. To start with, as Celma (2010) points out, user preferences for music are mostly tracked implicitly, by tracking their listening habits, rather than through rating systems. Indeed, we are interested in developing a Music RS for a network of devices, on which the songs listened to can be monitored. In addition, tracks can be consumed repeatedly in a short stretch of time, even continuously.

From the point of view of generating reviews, Music RSs need to recommend individual tracks, not full albums, e.g., see www.itunes.com. However, most available data consist of album reviews, not individual song reviews (see, e.g., www.amazon.com, www.epinions.com). References to and comments about a given song are scattered all over an album review. To generate a summary of reviews of individual songs in order to support a recommendation, those comments first need to be extracted from the album reviews. Even if a few music web sites, such as www.last.fm, allow the user to comment on individual songs, these comments are too short (a few words such as "awesome song" etc.) to count as a full review.

In this paper, we present SongRecommend, our prototype Music RS. We start by discussing our goals and presenting related work in Section 2: Our focus is on generating summaries of reviews to accompany recommendations, rather

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

It does have **"I'll Back You Up"**, and that's enough for me.
..... (269 words)
I think most of us know **"Ants Marching"**, one of the band's *biggest hits*, **"Tripping Billies"** and **"Recently"** were moderately popular *single* that have become live favorites, and **"Satellite"** was the first *Dave Matthews Ballad* (tm) that finished...
..... (150 words)
"One Sweet World" and **"Song that Jane"** (named for Dave's sister) are from a different live performance than the first four tracks.....
.... (169 words)
"Minarets" is a very *dark track* with an eastern sort of feel with *lyrics* that touch on the atrocities of holy wars.....(600 words)

Fig. 1. A sample review for the album 'Remember Two Things'.

than on generating the recommendations themselves. A general introduction to SongRecommend is provided 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. Section 5 is devoted to several evaluations that we performed. First, we conducted intrinsic evaluations of the extraction components, including a comparison of our feature extraction algorithm with the one in Hu and Liu (2004); and of the coverage of the summary, as compared to critic reviews extracted from www.rollingstone.com and other music sites. These intrinsic evaluations show that our algorithms perform very well, with most precision and recall results of over 90 percent. Second, we conducted an extrinsic evaluation of the system 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 The problem

The notion that people can be overwhelmed by information in this day and age has become commonplace. For many tasks, such as purchase of items and traveling and entertainment, users are faced with apparently endless sets of choices. Not surprisingly, many sites have emerged that thrive on collecting and presenting reviews to customers, such as TripAdvisor.com, reviews.cnet.com, and customersearch.com, just to name few of those available. Going beyond collecting and disseminating reviews, a few online recommendation systems have been developed, e.g., for books (www.amazon.com,www.bnn.com), movies (www.imdb.com), and music (www.itunes.com).

We contend that developing a recommendation system for songs presents additional challenges with respect to recommendation systems for products such as cameras, restaurants, and to a certain extent, books, and movies. Some of these challenges are derived from the fact that song reviews are mostly embedded within album reviews; some from the nature of songs considered as 'products'. We will illustrate these challenges by referring to the sample review of the album *Remember Two Things* by Dave Matthews Band, in Figure 1:

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

.... They complained that there was no **memory card** and....
I just bought this camera and it is so easy to use, it also has great **picture quality**. When you are not using it then it automatically powers off saving the life of the **battery**.....
You can't rate the quality of a product by what accessories it didn't come with....

Fig. 2. A sample review from Amazon.com for a digital camera.

(a) Eight songs (here marked in bold for reference) are mentioned in this excerpt. Each song needs to be identified each time it is referred to in the review. Whereas the title is the main identifier for individual tracks, titles are often abbreviated, and in different ways, even in the same review – e.g., see *Song that Jane* for *The Song that Jane Likes* in Figure 1. In addition, song titles need not be noun phrases (NPs) – e.g., *Ants Marching, Recently* – and hence they cannot be extracted with Named Entity Recognition techniques, which focus on noun phrases. Indeed, preliminary experiments we ran showed that NP extraction algorithms miss many occurrences of song titles. Also note that songs are not mentioned in any predefined order in a review, and most certainly the order of presentation will differ among reviews. For example, in Figure 1, *Back you up*, the ninth track in the album, is mentioned first (the order of mention for the other songs instead follows the order of tracks in the album).

(b) Comments on features of each individual song (*biggest hit, popular singles, dark track, lyrics)* are embedded and fragmented within an album review. It is necessary to correctly map each feature to the song it is attributed to.

(c) The set of features that reviewers focus on in song reviews is more heterogeneous than in product reviews, such as cameras or restaurants. This makes it harder to generate a coherent summary. For example, take reviews of cameras, such as the excerpt shown in Figure 2: features, such as *memory card*, *picture quality*, *battery*, belong to the same set across all cameras. On the other hand, features of songs, such as *Dave Matthews Ballad* in Figure 1, differ from one song to another even in the same album. For example, the artists mentioned for each song are often different. In song reviews, reviewers focus on both inherent features of songs such as lyrics, genre, and instruments that apply to any song, and on people (artist, lyricist, producer, etc.), unlike in product reviews, where manufacturer/designer are rarely mentioned. For this reason, song reviews are closer to reviews of other cultural objects such as movies or books (Zhuang *et al.* 2006) than to reviews of consumer products such as cameras. However, book or movie reviews still focus on the individual book or movie, in contrast to song reviews that are mostly embedded within album reviews, as we have just discussed.

The premise our work is based on is that song reviews are an appropriate piece of information to provide to support song recommendations. To start with, note that research shows that users trust recommendation systems more if recommendations are accompanied by explanations (Tintarev and Masthoff 2007). The question is,

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

what sort of explanations are most useful? We ran a small pilot study, in which three users were presented with a preliminary version of our system interface (see Figure 3), and provided with different types of support for the recommendations: reviews of the recommended song, blogs related to the song, videos of different versions of the song, news articles about the artist or the song, and local events related to the artist or song or album. Users were asked to rate the importance of each type of support on a Likert scale from 1 to 5 (1 = not important, 5 = most important). Reviews were ranked the highest at 3.67, followed by events at 3, blogs at 2.67, videos at 2.33, and news at 1. Not only were reviews the highest ranked on average, they were the most consistently ranked across the three users, whereas the other items (other than news) showed large differences in rankings (no item was ranked as 5 by any user). Hence, we focused on song reviews.

We will now further illustrate the points that we made by contrasting song review summarization with three relevant areas of research: feature extraction, summarization, and development of recommendation systems.

2.1 Related work

2.1.1 Feature extraction

To generate summaries of customer reviews, a necessary step is to first extract the most important features that customers focus on. One of the first, and influential, papers in this regard is Hu and Liu (2004). In this work, associative mining techniques are used 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. We use Hu and Liu's (2004) approach as a benchmark to evaluate the feature extraction component of our work (see Section 5.1). The feature extraction component in Carenini *et al.* (2006), to be discussed later, is based on and enhances the approach in Hu and Liu (2004). Rather than extracting features *per se*, another work, such as Gamon *et al.* (2005), focuses on identifying polarity of opinion terms and on clustering sentences as a way to extract topics. Feature extraction to summarize reviews on objects different from products, such as restaurants (Higashinaka, Prasad and Walker 2006; Nguyen *et al.* 2007), or movies (Zhuang *et al.* 2006), has also been tackled, although not as extensively. Higashinaka *et al.* (2006) approach the problem from a different angle, since they focus on extracting domain-specific ontologies from user reviews. We are aware of only one piece of work that focuses on mining music reviews (Downie and Hu 2006). This study is mainly concerned with identifying descriptive term patterns to differentiate positive from negative reviews, but not on summarizing the reviews.

All the works discussed above use a combination of machine learning and Natural Language Processing (NLP) techniques, where the focus is most often on the machine learning aspects rather than on the Natural Language Processing techniques. In our work, we reverse the emphasis.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

2.1.2 Summarization

Most works on product reviews focus 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. Indeed, the summarization framework described in this paper is illustrative of multi-document summarization (Barzilay and McKeown 2005; Soubbotin and Soubbotin 2005; Nastase 2008), since it generates summaries of songs from multiple album reviews. The difference with most works in multi-document summarization is that the latter does not focus on evaluative text. An exception is Carenini *et al.* (2006), where two approaches to summarization of evaluative texts are compared: Summarizing extracted sentences, where sentences are ranked according to various aspects of the features they contain; and generating new text based on algorithms for feature selection and for computing distribution of opinions, and a full natural language generation system. A user evaluation of the two approaches found that both were significantly better than a baseline standard approach to multi-document summarization, and that each presented different strengths and weaknesses. At least at an abstract level, our approach can be seen as a hybrid of the two approaches in Carenini *et al.* (2006): We do extract sentences based on the features they contain and on polarity of opinion, however, we regenerate them via aggregation and surface generation, as opposed to the full regeneration of Carenini *et al.* (2006) or Saggion and Lapalme (2002). Because of this, we follow Xie *et al.* (2008) in describing our summarization approach as *quasi-abstractive*: It goes beyond extractive summarization in that sentences are not necessarily included in the same form in which they were extracted; however, it falls short of fully abstractive summarization in which the summary includes components that did not appear in the original document.

2.1.3 Recommendation Systems

Whereas the main focus of the work described in this paper is to summarize reviews of songs after they have been recommended, and not on how such recommendations are arrived at, our prototype should be ultimately embedded in a full-fledged Music RS. Here we briefly review some of the main approaches to developing RSs. The first three methods have reached maturity, hence examples of commercial products developed using these techniques are available.

- *Collaborative or social filtering*: The main idea behind this method is that the user gives feedback to the system, for example, by rating some items; based on the feedback provided by other users, the system gives recommendations. Some drawbacks of this method include the Cold-Start Problem (when there are few users to provide feedback), bias toward popular items (only popular items are recommended), and the early-rater problem (new items cannot be recommended until users rate them). Moreover, this approach does not take the descriptions or the characteristics of the items into consideration. Examples of this approach are *Tapestry* (Goldberg *et al.* 1992) and *MovieLens* (Miller *et al.* 2003).

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

- *Content-based filtering*: The information about the items is collected and then based on user preferences, the system predicts what the user might like. Though this approach solves the early-rater problem of the collaborative filtering method, the remaining problems still persist. Examples are *Music-Surfer* (Cano, Koppenberger and Wack 2005) and *PRES* (Van Meteren and Van Someren 2000).
- *Demographic filtering*: These RSs aim at categorizing users based on their demographic information (age, gender, etc.) and recommend services accordingly. More precisely, demographic information is used to identify the types of users that like similar services. Examples are *Grundy* (Rich 1979) and (Le Roux *et al.* 2007).
- *Context-based filtering*: These RSs take into account external information, such as reviews and editorial data, together with user preferences and the item information. The key goal of context-aware systems is to provide a user with relevant information based on his current context. An example is *COMPASS* (Van Setten, Pokraev and Koolwaaij 2004), and in the music arena, *FOAFing the music* (Celma 2010).

The recommendation component proper of SongRecommend, our Music RS, is very simple at the moment. In principle, a more realistic version of this component could be developed according to any of the methods just listed (assuming the necessary resources exist). We believe that a hybrid of content- and context-based filtering would be the most appropriate version. SongRecommend monitors songs listened by the user on several portable and non-portable music devices and recommends additional songs, which may either exist in the user's collection or are downloaded from the Internet; in addition, we have already developed a prototype to learn user preferences, although it was not used in the system that we describe in this paper (see Section 6 for further details). At the same time, the review summaries SongRecommend generates could be used to rank the recommendations as well.

3 SongRecommend: our prototype music RS

Figure 3 shows the architecture of our system. As shown in the figure, personal music devices are monitored – most personal electronic devices follow a Universal Plug and Play (UPNP) protocol (UPN 2008), and can be used in a simple network. The *User Action Collector* collects all the user's actions on the device and identifies the songs listened by the user on different devices. Related songs for each of the songs listened by the user are obtained from the web by the Expander. The recommendation algorithm is run to present the users with recommendations in the interface of the Recommendation System (Section 3.1). Along with the information about the song, a summary of the song reviews is generated by the summarizer (the window at the bottom of Figure 4). First, the extraction module obtains features of the song from the album reviews; second, the summarization module presents them as a coherent summary. Details about the extraction and summarization modules, which are the main focus of this research, are provided in Section 4. For further details on SongRecommend, the reader is referred to Tata (2010).

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

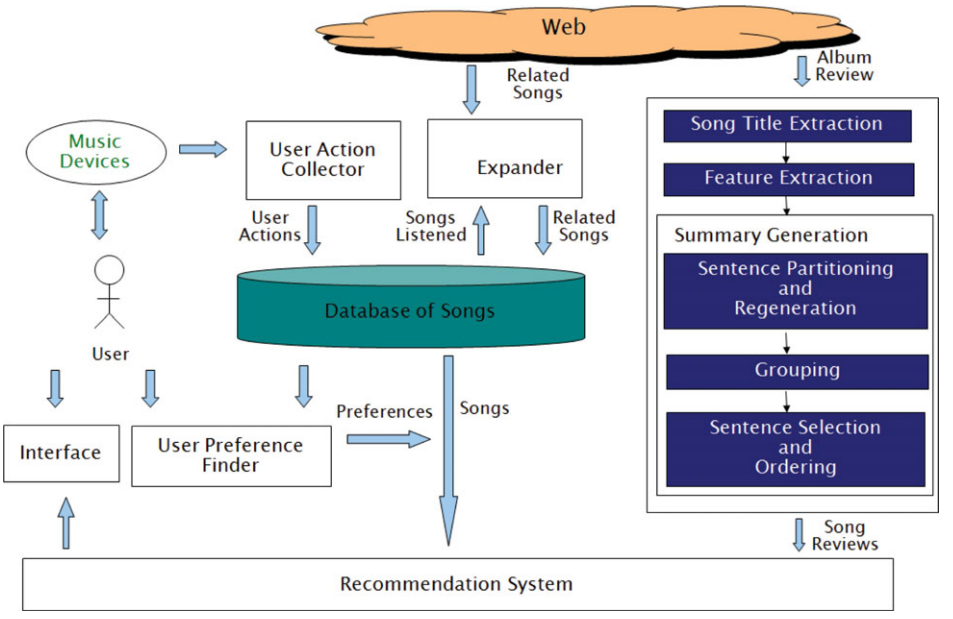


Fig. 3. (Colour online) Architecture of SongRecommend.

3.1 User interface

Figure 4 shows the interface of our prototype Music RS. It is a simple interface, dictated by our focus on the summarization process. It has four windows in the top part, and a bottom, pop-up window that displays details about the recommended song, or the review summary. Moving from left to right, the four windows on top display the following information:

(1) The top leftmost window shows the devices on which the user listens the songs. These devices are monitored with a UPNP control point, which receives messages from the devices. Based on these messages, user activities, including the metadata of the song she/he chooses, are logged.
(2) The second window from the left shows the list of songs available on the device selected in Window 1.
(3) The third window displays more information about the song the user has chosen in Window 2.
(4) At this point, the Expander (see Figure 3) identifies songs related to the chosen song 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. Among those downloaded songs, we identify and rank the top twenty-five songs which belong to the same cluster and are closest to the given song – we use SimpleKMeans Clustering (Mitchell 1997). Closeness between two songs in a cluster is measured as the number of attributes (album, artist, etc.) of the songs that match. These top twenty-five recommendations are shown in the rightmost top window (in Figure 4, only the top recommendation is shown).

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

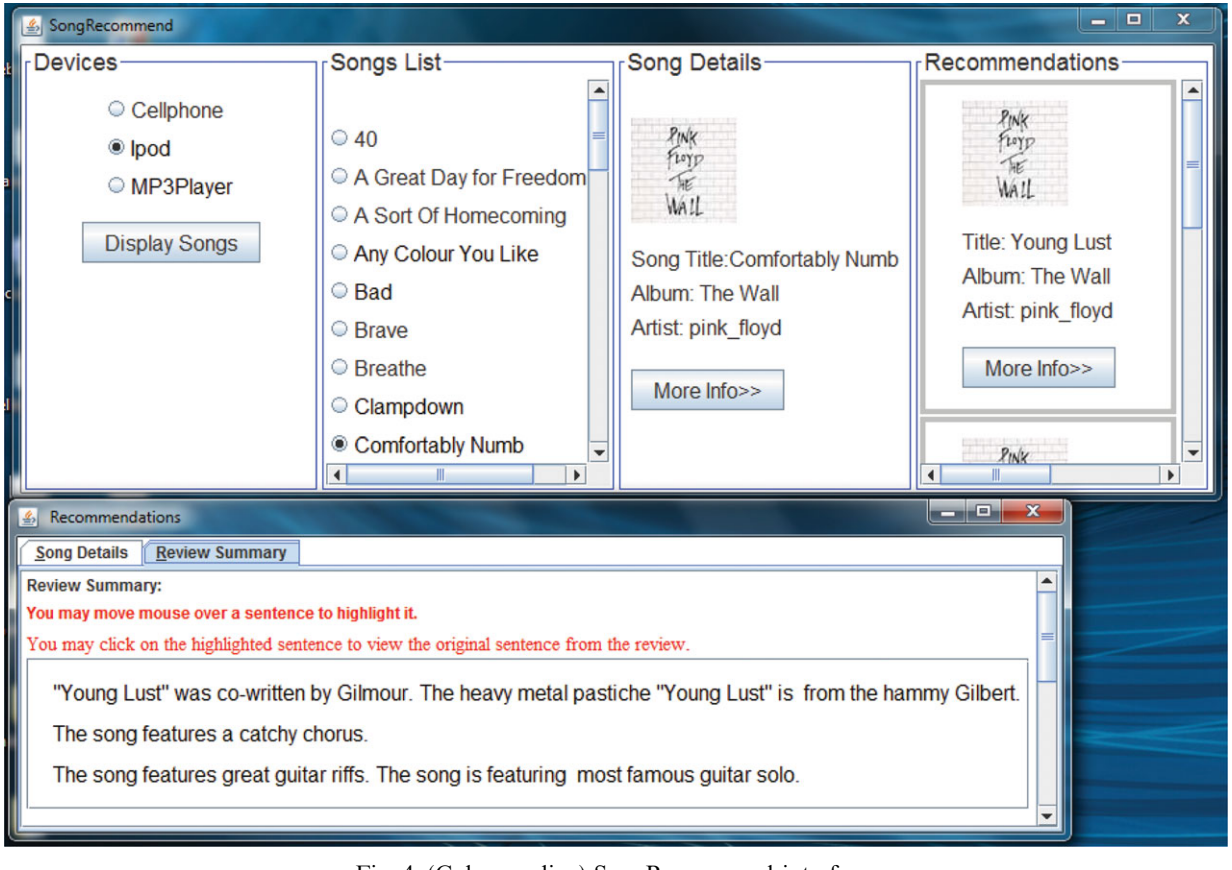


Fig. 4. (Colour online) SongRecommend interface.

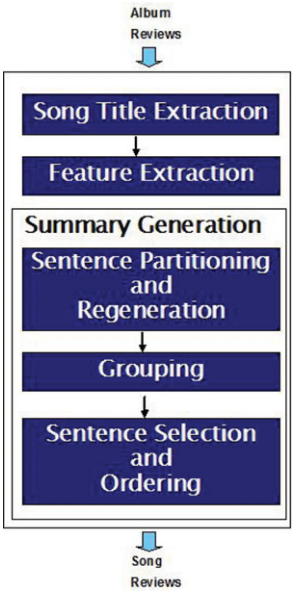


Fig. 5. (Colour online) Summarization pipeline.

If 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 system is a pipeline (shown on the right in Figure 3, and repeated in Figure 5) of the following five steps:

(1) Identification of song title occurrences in an album review.
(2) For each song, identification of sentences containing features.
(3) For each sentence, partitioning and regeneration of the sentence into several simpler sentences, each containing only one feature.
(4) Grouping regenerated sentences based on their polarity and the (sub)features they contain (we will later explain the sub-features).
(5) Selection of one regenerated sentence from each group, and ordering of the selected sentences.

The first two tasks pertain to information extraction, the last three to repackaging the information and generating a coherent 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 via both the intrinsic evaluation and the user study, to be discussed in Section 5.

Our corpus was opportunistically collected from two web sites, www.amazon.com and www.epinions.com, to make sure that our algorithm does not have any structural

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

bias depending on the structure of the reviews. We selected fifty reviews each for twenty-seven albums, for a total of 1,350 album reviews. These twenty-seven albums contain a total of 391 songs. As much as possible, we tried to collect an equal number of reviews from both web sites for each album. However, for certain albums, if one web site did not have twenty-five reviews, we added reviews from the other web site, up to the desired fifty reviews. The reviews extracted from those two web sites are the most recent reviews given by users. We then manually inspected this initial set of 1,350 album reviews for length and relevance, since some reviews included in the original set were too short, e.g., they only contained one sentence, or did not contain anything related to songs at all. Since we needed to assemble a gold standard corpus, automatic filtering of irrelevant reviews was not a viable option: First, a corpus would have been necessary to train the automatic filter, but as we noted, no corpus existed when we set out on this project; second, automatic filtering would have most likely resulted in several false positives, reducing the size of the corpus; and in several false negatives, affecting the cleanliness of the corpus.

After manual inspection for length and relevance, if for one album, the fifty reviews initially collected contain N short and/or irrelevant reviews, we attempt to substitute as many of these short or irrelevant reviews as possible with the '*Most helpful reviews*' or the '*Most critical reviews*' that are sometimes available on the amazon and epinions web sites. Hence, for each album, the final set of fifty reviews may include $K = N - M$ short or unrelated reviews. How large is K depends on how many substantial reviews, S, are available for that album to start with: if $S = 50$, $N = 0$, and hence $K = 0$; if $S < 50$, on how many (M) '*Most helpful review*' and/or '*Most critical reviews*' are available.

Along with the reviews, we also extracted the titles of all the songs in that album. These titles are required for the song title extraction algorithm. Figure 6 shows the distribution of the reviews based on number of words. Reviews varied in size from twenty-two words to 3,779 words (about seven pages). On average, the reviews contained about 478 words, that is, a little more than one page of text. Most of the reviews contain less than 500 words.

Fifty randomly chosen reviews were used for development. We used the reviews as we had collected them, and did not clean them in any way, for example, we did not correct spelling mistakes – Figure 7 shows a review with spelling mistakes (highlighted in boldface). The entire corpus was annotated for song titles and song features. Feature annotation consists of marking a phrase as a feature and identifying the song to which the feature is attributed. We did not provide the annotators with a predefined inventory of features; what counts as features of songs emerged from the annotation, since annotators were asked to annotate for phrases which contain 'any song-related term or terms spoken in the context of a song'. Further, they were given about five positive and five negative examples of features. Figure 8 shows annotations for the excerpt from Figure 1: *biggest hits, singles*, *Dave Matthews Ballad*, *dark track*, and *lyrics* have been marked as features. The feature annotation shows that features are attributed to songs: For example, the feature *singles* is attributed to songs 2 and 3 (*Tripping Billies* and *Recently*).

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

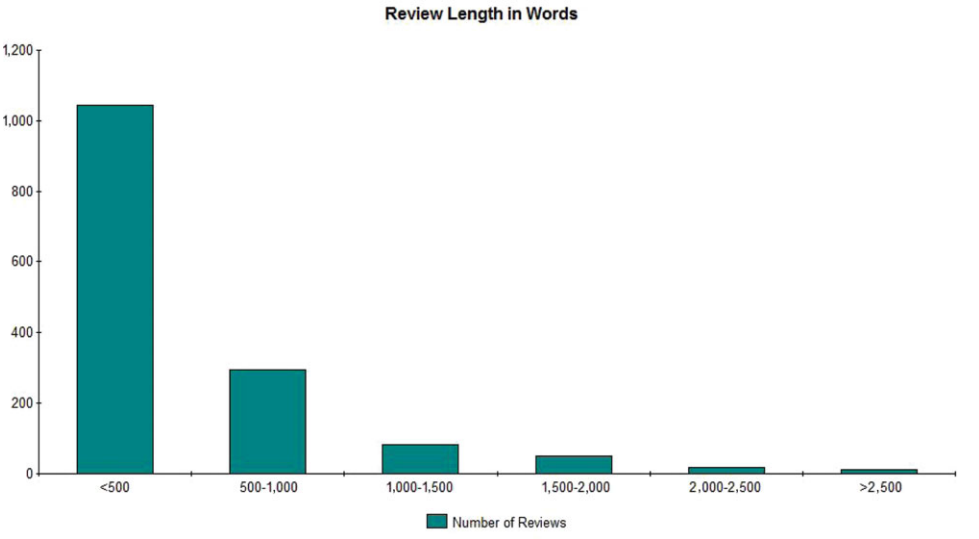


Fig. 6. (Colour online) Length of reviews in number of words.

He has been dubbed as one of the greatest guitarists of all time yet there is something about Jimi that **seperates** him from the rest.....
I can only say that is **definetly** essential Hendrix material ....
After that is one of my favourites I DON'T WANT TO LIVE TODAY followed by the unique WIND CRIES MARY....
.... Why? because music has not been the same since!

Fig. 7. An excerpt from a sample review containing spelling mistakes.

To check inter-coder agreement, our two annotators were asked to annotate a set of ten randomly chosen reviews. These reviews contained 4,500 noun phrases. Two Kappa values (Carletta 1996) were computed, one for song IDs and another for features. 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. In both cases, the Kappa coefficient was 0.9, which is considered to be an excellent match. Hence, each of the remaining reviews was annotated by one annotator only.

We will now turn to describing the component tasks of our approach.

4.1 Title extraction

Song identification is the first step toward summarization of reviews. Unless songs are identified, the features cannot be made sense of, since users comment on different features for different songs, or features of the album as a whole. 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

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

..... I think most of us know <**songid=1**>**"Ants Marching**</**song**>**"**, one of the band's <**feature id=(1)**>**biggest hits**</**feature**>,<**songid=2**>**"Tripping Billies**</**song**>**"** and <**songid=3**>**"Recently"**</**song**> were moderately popular <**feature id=(2,3)**>**singles**</**feature**> that have become live favorites, and <**songid=4**>**"Satellite"**</**song**> was the first <**feature id=(4)**>**Dave Matthews Ballad**</**feature**> (tm) that finished...
.....
<**songid=5**>**"One Sweet World"**</**song**> and <**songid=6**>**"Song that Jane"**</**song**> (named for Dave's sister) are from a different live performance than the first four tracks.....
....
<**songid=7**>**"Minarets"**</**song**> is a very <**feature id=(7)**>**dark track**</**feature**> with an eastern sort of feel with <**feature id=(7)**>**lyrics**</**feature**> that touch on the atrocities of holy wars.....

Fig. 8. A sample-annotated review.

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.*

At first approximation, extracting song titles could be approached with Named Entity Recognition (NER) techniques. However, as we noted in the Introduction, NER techniques focus on NP extraction. Since song titles range the spectrum of syntactic phrases (verb phrases, adverbs, etc), NP extraction algorithms miss many song title occurrences as shown by preliminary experiments we ran. Others have noted that NP extraction algorithms fail in the music domain, and have proposed different approaches, for example, the algorithm from Schedl *et al.* (2007), which extracts band names from web sites. This algorithm is based on *n*-grams, and forms the basis for our approach to song title extraction. We proceed album by album. Figure 9 shows the sequence of steps for song title extraction from the reviews. 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. Each word in the lexicon is associated with all the IDs of the songs in whose titles it appears. We also segment the reviews into sentences via sentence boundary detection.

All 1, 2, 3, 4-grams for each sentence in the review are generated (the upper-bound 4 was determined experimentally, as we discuss below). 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. Including those *n*-grams with an edit distance of one will help to identify titles with spelling mistakes. Second, if higher and lower order *n*-grams overlap at the same position in the same sentence, lower order *n*-grams are filtered out. While finding the lower order *n*-gram we take into consideration the position of the words of the *n*-gram in the sentence. Only those lower order *n*-grams that represent the same word position as that of the higher order *n*-grams are filtered. For example, if given the *n*-gram [*Song*, *That*, *Jane*, *Likes*] we filter out all the other *n*-grams (< 4) that occur in positions overlapping with the longer *n*-gram, in the same sentence –

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

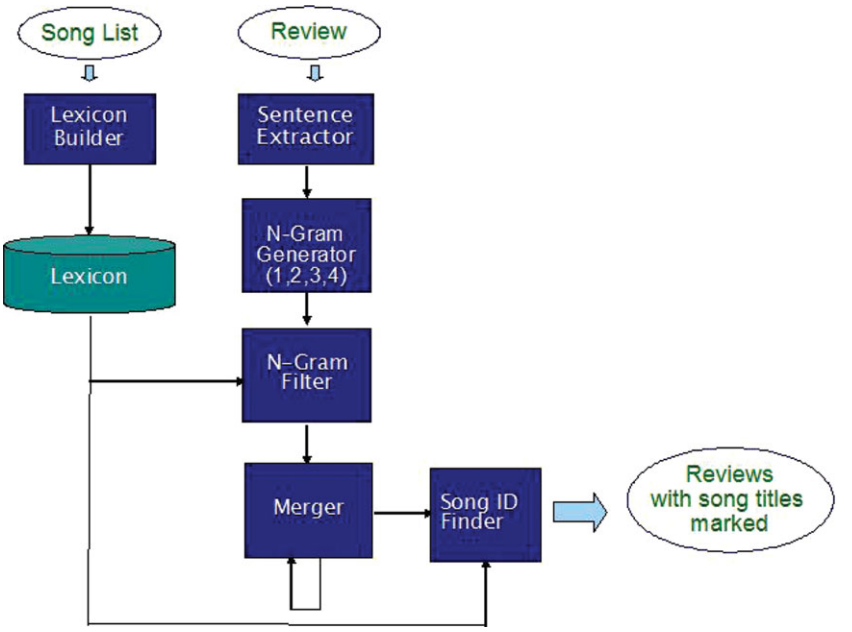


Fig. 9. (Colour online) Flow chart for song title extraction algorithm.

for example, [*Song*], [*That*, *Jane*, *Likes*]. Third, the remaining 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 results 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 and there is at least one content word that distinguishes every song title from others in the same album. The filter in step 4, which eliminates n-grams that are neither surrounded by quotes nor capitalized, may appear too restrictive, since the writing in these reviews is informal. Nonetheless, reviewers appear to adhere to these unspoken conventions, as shown by the recall we obtain on song title extraction, 99.2 percent (see Section 5.1).

As mentioned, we established $n = 4$ as the upper bound for our n-grams based on observations and a small experiment. First, we observed that song titles longer than three or four words are less frequent. Using as an example three albums, each with ten tracks, by three different artists: in *Remember Two Things* (Dave Matthews Band), only two songs have titles longer than three words (one has four words and the other, five); likewise, *Abbey Road* (Beatles) has only two songs whose title is longer than three words (one has four words and the other, six); *War* (U2) has one song whose title is longer than three words (five words total). Second, we observed that even for longer titles, most reviewers refer to them in an abbreviated form (see, e.g., *The song that Jane*, for *The song that Jane likes* in Figure 1). These observations

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

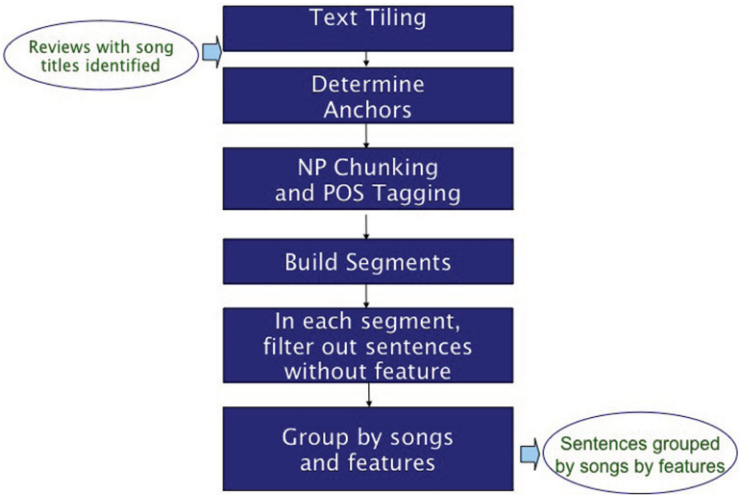


Fig. 10. (Colour online) Flow chart for feature extraction algorithm.

suggest that an upper bound of $n = 4$ should be sufficient to capture the vast majority of references to song titles. The question is whether we could lower it to, e.g., $n = 3$. However, the lower the bound on n, the higher the number of mergings of sequential n-grams the algorithm needs to perform (this is the most expensive step in the algorithm). In a small experiment on four reviews each for *Abbey Road* and *Rubber Soul*, we found that using $n = 3$ as the upper-bound on n-gram length generated sixty-seven mergings, whereas using $n = 4$ only thirty-eight mergings were necessary. Hence, we converged on $n = 4$.

4.2 Feature extraction

After song titles are identified, each sentence that contains one song title can be used as an anchor to identify *segments* of texts that talk about that specific song, and to extract the feature(s) that the pertinent text segment discusses. This task roughly corresponds to identifying the flow of topics in a review. In principle, it could benefit from sophisticated topic models or from reference resolution algorithms. However, given the still lacking performance of state-of-the-art algorithms for these tasks, we chose a simpler method. We devised an approach that combines TextTiling (Hearst 1994) and domain heuristics. The TextTiling algorithm divides the text into coherent discourse units to describe the subtopic structure of the given text. TextTiling assumes that a set of words is in use while a certain subtopic is being discussed, and when that subtopic changes, a significant proportion of the vocabulary changes as well.

Figure 10 shows in more detail the sequence of steps performed during the feature extraction process. First, the reviews (pre-processed by demarcating sentence boundaries as discussed above) are divided into discourse units (DUs) using the TextTiling algorithm. Let $[DU_1, DU_2, \ldots, DU_k]$ be the units obtained. The segments that contain potential features of a song are identified using the following heuristics:

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

(1) *Determine anchors*: Include DU_i in segment s_j if it contains a song title (as identified by the title extraction algorithm). These DUs are more likely to contain features of songs as they are composed of the sentences surrounding the song title.
(2) *Build segments*: Include DU_{i+1} in segment s_j if DU_i is included in s_j and DU_{i+1} contains one or more *feature terms*.

Building of segments rests on the notion of 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 these. We consider as feature terms those noun phrases whose synonyms, direct hypernym or direct hyponym, or the definitions of any of these, contain the terms 'music' or 'song', or any form of these words, such as 'musical', 'songs', etc, for at least one sense of the noun. Feature terms do not include the words 'music' or 'song' *per se*, or the artist/band/album name as these are likely to occur across a whole album review.

We just mentioned that feature terms include only noun phrases. The exclusion of other grammatical phrases rests on the following analysis. We randomly chose ten reviews from the corpus; those ten reviews contained 1,473 noun phrases and seventy-eight features. Of these seventy-eight features, only two (3.4%) are not noun phrases. One of those two is *long*, in the sentence, *'Clocking' is not only long, it is also one of the most difficult songs.* Those non-NP features could potentially be identified with natural language understanding techniques, for example, linking an adjective, such as *long*, to a noun such as *length*. However, given the heterogeneity of those non-NP features, the inherent limitations of the current state-of-the-art Natural Language Processing techniques and, above all, the small number of non-NP features we found in our data, we only took into account features of a song that are nouns or noun phrases. This is consistent with most of the works that performs explicit feature extraction from product reviews such as Hu and Liu (2004), Popescu and Etzioni (2005), Carenini *et al.* (2006), Higashinaka *et al.* 2006, and Zhuang *et al.* (2006) – work, such as Jin, Ho and Srihari (2009), includes few rules aimed at features not expressed as noun phrases; however, their evaluation does not assess the impact of these rules on their system's performance.[1] Likewise, the comments that the user provides on these features, and that we analyze, can only be adjectives or noun phrases.

The choice of limiting feature terms to noun phrases explains why Figure 10 includes a step concerned with part-of-speech (POS) tagging and NP chunking. We use Ramshaw's POS tagger and Noun-Phrase Chunker (Ramshaw and Marcus 1995).[2] Once feature terms have been identified, sentences that do not contain any feature terms are filtered out. At this point, the algorithm has identified segments of the album review such that each segment pertains to one song, each sentence in that segment contains a feature term, and all feature terms in those sentences are taken to be features of the song described by that segment. Since our interest is in

[1] Much work on opinion mining, such as Gamon *et al.* (2005) and Nguyen *et al.* (2007), does not extract explicit product features to start with.
[2] As re-implemented in Java by Mark Greenwood.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

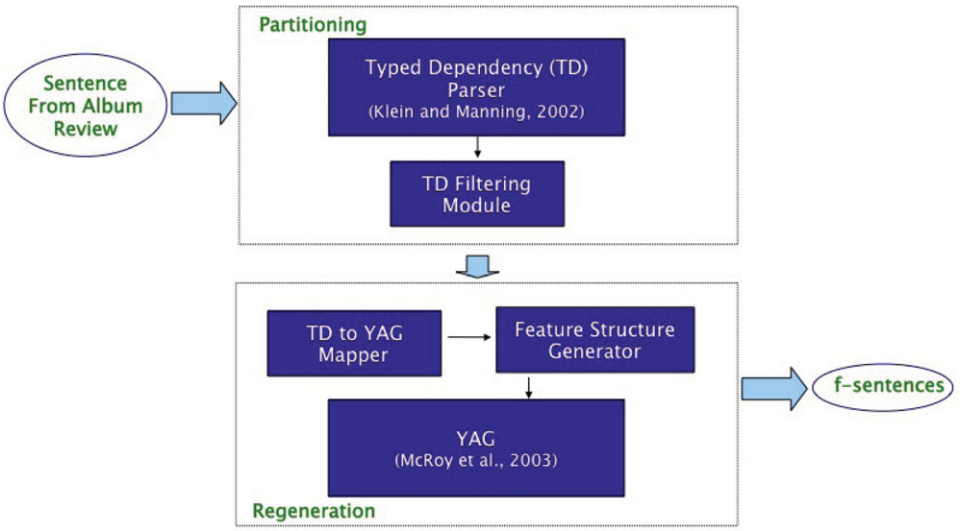


Fig. 11. (Colour online) Flowchart for sentence partition and regeneration.

extracting appropriate individual sentences for the summary, segments will not be used further; however, the sentences contained in those segments are then grouped based on the feature term, and on the ID of the song to which the feature term corresponds to. If a sentence contains more than one feature term, it will appear in as many groups as feature terms it contains. Duplicates will be eliminated as a by-product of sentence partitioning, to which we turn now.

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, '*The slightly awkward wording of the lyrics is endearing, and Tim Reynolds' guitar is impressive*' discusses both features, *lyrics* and *guitar*. Only a portion of the sentence describes the *guitar*. This sentence can thus be divided into two individual sentences:

(1) '*The slightly awkward wording of the lyrics is endearing*'.
(2) '*Tim Reynolds' guitar is impressive*'.

Only the second sentence is relevant to the feature *guitar*, whereas the first sentence is relevant to the feature *lyrics*. 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.

Figure 11 depicts the entire process of sentence partitioning and regeneration. First, to identify portions of the sentence relevant to the single feature f, we use

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

det(line-5, A-1)
amod(line-5, slow-2)
amod(line-5, methodical-3)
nn(line-5, drum-4)
nsubj(begins-6, line-5)
det(song-8, the-7)
dobj(begins-6, song,-8)
nsubj(matter-10, that-9)
rcmod(song-8, matter-10)
det(strongly-13, this-11)
nn(strongly-13, album-12)
dobj(matter-10, strongly-13)

Fig. 12. Typed dependencies, to be filtered to generate f-sentence.

the Stanford Typed Dependency Parser (Klein and Manning 2002; de Marneffe 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 nouns, adjectives, and verbs that are related to it in the sentence, either directly or recursively, but without including *relative clause modifier* (*rcmod*) dependencies on modifiers. These nouns, adjectives, and verbs become the components of the new f-sentence.

Example: Consider the following sentence: *A slow, methodical drum line begins the song, that matter this album strongly*.[3] The typed dependency parse for this sentence is shown in Figure 12. Because of the filtering module, only the dependencies in bold (corresponding to the words [*a, slow, methodical, drum, line, begins, the, song*]) are identified as components of the f-sentence for *drum line* – the *rcmod* dependency on *song* and all its attendant dependencies are filtered out.

To generate a grammatical sentence from the words identified as components of the f-sentence, we may need to perform adjustments to their form, for example, to the number of nouns and verbs. This is a natural language generation task, specifically, sentence realization. We use (McRoy, Channarukul and Ali 2003), a sentence realizer that uses templates augmented with rules to express text structures corresponding to fragments of the target language. In each YAG template, slots represent variables to be filled with values, whereas rules are declarative statements that define how inputs to the template should be realized as text. An example of the instantiated *clause* template, the root of the top–down realization process, is shown in Figure 13 (the rules that have been fired to fill it are not shown). The dependencies obtained from the Typed Dependency Parser, such as *subject* and *object*, identify the slots and the template each slot follows, since slots can be templates themselves – see the *subject* and *dobject* slots in Figure 13. We developed a mapping between the Stanford Typed Dependencies and the slots that YAG needs (see Tata 2010, Table V). The words in the dependency fill the slot.

[3] This sentence reads somewhat awkward. It appears in its original form, since, as we mentioned, we did not clean our corpus in any way.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

Feature: *drum line*

```
(null ((template "clause")(sentence "yes")(process-type "MATERIAL")
      (subject ((template "noun-phrase")(head "drum line")(definite "")
               (np-type "COMMON")(determiner "A")(describer "slow methodical")))
      (dobject ((template "noun-phrase")(head "song")(determiner "the")(definite "")
               (np-type "COMMON")(describer "")))
      (tense "present")(process "begin")))
```

Fig. 13. YAG template to generate '*A slow methodical drum line begins the song*'.

1. **"Ants Marching"** is one of the band's biggest hits.
2. **"Tripping Billies"** is moderately popular single.
3. **"Recently"** is moderately popular single.
4. **"Satellite"** was the first Dave Matthews Ballad.
5. **"One Sweet World"** is from a different live performance.
6. **"Song That Jane"** is from a different live performance.
7. **"Minarets"** is a very dark track with an eastern sort of feel.
8. **"Minarets"** has lyrics.

Fig. 14. f-sentences corresponding to Figure 1.

We use a morphological tool (Minnen, Carroll and Pearce 2000) to obtain the base form from the original verb or noun so that YAG can generate grammatical sentences. YAG regenerates as many f-sentences from the original sentence as many feature terms are contained in it.

By the end of the partitioning and regeneration steps, 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. Figure 14 shows the regenerated review from Figure 1.

4.4 Grouping

f-sentences are further grouped, by sub-feature and polarity. Sub-features are noun phrases that include a feature term, but define either sub-concepts or properties of the concept described by the feature term. Consider the following f-sentences for the feature *violin*:

(a) *The violin solo is all well done.*
(b) *It has a nice violin solo by Boyd.*
(c) *The lead violin is noteworthy.*

The first and second sentences talk about the *violin solo* and the third sentence about the *lead violin*. This step will create two subgroups, with sentence (a) and (b) in one group and sentence (c) in another. We 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 15. For two f-sentences, if the Longest Common N-Gram (LCN) is a phrase containing the feature term under

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

Let $[f_x\text{-}s_1, f_x\text{-}s_2, \ldots f_x\text{-}s_n]$ 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 subgroup* sub_f^k.

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 feature f-group.*

Fig. 15. Grouping sentences by sub-features.

consideration, but is not the feature term alone, then the two f-sentences belong to the same subgroup sub_f^k, otherwise they are in different subgroups. If the LCN of an f-sentence with all other f-sentences is the feature term alone, then the f-sentence belongs to the feature f-group.

When the procedure described in Figure 15 is applied to the three sentences above, step 2 will group (a) and (b) together since *violin solo* properly contains the feature term *violin*.

In step 3, the LCN between (c) and all other sentences contains only the feature term *violin*; hence, sentence (c) is left on its own. Note that steps 2 and 3 ensure that, among all the possible LCNs 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 or sub-feature. To generate a coherent summary that mentions conflicting opinions, we need to subdivide f-sentences in each group or subgroup $sub_{f,i}$ according to polarity.

We use SentiWordNet (Esuli and Sebastiani 2006), an extension of WordNet where each sense of a word (each synset from WordNet) 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. Although this is a simplification, it is supported by findings in the literature that show that adjectives are considered among the core clues to polarity, even if other types of words also contribute to the perceived polarity of a sentence. First, adjectives correlate with the subjectivity of a sentence (Hatzivassiloglou and McKeown 1997; Bruce and Wiebe 1999; Wiebe, Bruce and O'Hara 1999), the first step toward determining its orientation (objective versus subjective, with subjective comprising positive or negative). Second, adjectives strongly contribute to the polarity of a sentence: Esuli and Sebastiani (2006) found that adjective and adverb synsets are evaluated as at least partially subjective three times more frequently than verb or name synsets. Esuli and Sebastiani (2006) note that *[this] seems to indicate that, in natural language, opinionated content is most often carried by parts of speech used as modifiers (i.e., adverbs, adjectives) rather than parts of speech used as heads (i.e., verbs, nouns)*. Third, the added value for polarity determination of other classes of

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

words beyond adjectives is still a subject of research (Benamara *et al.* 2007; Ding, Liu and Yu 2008).

Since there are a number of senses for each word, an adjective a_i in a sentence is scored as a normalized weighted score for each orientation of the adjective (recall that SentiWordNet includes probability for each orientation – neutral, positive, and negative – for each sense of a word in WordNet). For each a_i, we compute three scores, positive, as shown in Formula 1, negative, and objective, which are computed analogously:

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

where a_i is the ith adjective in the sentence: for each of its senses j, $freq_j$ is the frequency count of the jth sense of a_i as given by Wordnet,[4] and pos_j is the positive score of the jth sense of a_i, as given by SentiWordnet. For an f-sentence, three scores will be computed, as the sum of the corresponding scores (positive, negative, and objective) of all the adjectives in the sentence. The polarity of the sentence is determined by the maximum of these three scores. Figure 16 shows an example of calculating the polarity of a sentence that contains two adjectives *beautiful, uneven* with different overall polarity (positive and negative respectively) – as shown in Figure 16, the polarity of the sentence is positive. Saggion and Funk (2010) propose a computation similar to ours, where the polarity of a sentence results from aggregating the SentiWordNet scores of the words in the sentence. However, Saggion and Funk (2010) take into account every word in the sentence, as opposed to only adjectives as we do here. In addition, Saggion and Funk (2010) use sentence polarity as a feature to assess the polarity of extended texts (mostly reviews), whereas we are interested in the polarity of individual sentences only.

4.5 Selection and ordering

We are now ready to undertake the last step in our summarization pipeline (see Figure 5): 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 sub-features, and one polarity type (positive, negative, and 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 sub-feature, 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 no user commented on noticing inconsistencies in the reviews (see Section 5.3).

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

[4] In Wordnet, the frequency counts for senses (synsets) corresponds to the number of times, possibly zero, a specific sense is tagged in the SemCor corpus (Miller *et al.* 1994).

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

Example: *The guitar solo is beautiful even if uneven*
Adjectives: *beautiful, uneven*

SentiWordnet Scores of *beautiful*:

Sense 1 (frequency = 25):
positive = 0.75, negative = 0.25 , objective = 0
Sense 2 (frequency = 0):
positive = 0.625, negative = 0.375, objective = 0

Polarity Scores Calculation:
positive(beautiful) = $\frac{25\times0.75+0\times0.625}{(25)} = 0.75$
negative(beautiful) = $\frac{25\times0.25+0\times0.375}{(25)} = 0.25$
objective(beautiful) = $\frac{25\times0+0\times0}{(25)} = 0.0$

SentiWordnet Scores of *uneven*:

Sense 1 (frequency = 4):
positive = 0.125, negative = 0.5 , objective = 0.375
Sense 2 (frequency = 1):
positive = 0, negative = 0.625, objective = 0.375
Sense 3 (frequency = 0):
positive = 0.5, negative = 0.125, objective = 0.375
Sense 4 (frequency = 0):
positive = 0.125, negative = 0.5, objective = 0.375
Sense 5 (frequency = 0):
positive = 0.625, negative = 0.125, objective = 0.25

Polarity Scores Calculation:
positive(uneven) = $\frac{4\times0.125+1\times0.0+0\times0.5+0\times0.125+0\times0.625}{(4+1+0+0+0)} = 0.125$
negative(uneven) = $\frac{4\times0.5+1\times0.625+0\times0.125+0\times0.5+0\times0.125}{(4+1+0+0+0)} = 0.525$
objective(uneven) = $\frac{4\times0.375+1\times0.375+0\times0.375+0\times0.375+0\times0.25}{(4+1+0+0+0)} = 0.375$

Max (positive(beautiful)+positive(uneven), negative(beautiful)+negative(uneven), objective(beautiful)+objective(uneven)) = Max (0.825, 0.775, 0.375)

Fig. 16. Polarity calculation.

subgroup is chosen as the representative for that subgroup. If multiple sentences have the lowest score in a subgroup, one sentence is selected randomly, since they are equivalent from the point of view of which feature they mention, and of their polarity. 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. Figure 17 shows an example list of f-sentences to be ordered to get the final summary.

We experimented with three different strategies for ordering the sentences in the summary:

(a) Noun-Phrase Centrality (NP-Centrality) (Xie 2006; Xie, Di Eugenio and Nelson 2008)
(b) Wordnet (Fellbaum 1998)
(c) Music Ontology (Raimond *et al.* 2010)

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

1. The song features pulsing drums by Carter.
2. The haunting, middle-eastern tinged Minarets sounds like a pre-cursor.
3. Minarets has become a rare performance.
4. It has an interesting sound.
5. Minarets is a very dark track.
6. The tone does not fit the rest of the album.
7. The song features the eerie strings.
8. A good saxophone riff sounds a little choppy.
9. The lyrics are pretty cool.
10. Dave uses patented hick-up vocals.

Fig. 17. f-sentences to be ordered.

In the development of our approach, we systematically moved from one strategy to the next (in the order given) by informally evaluating the summaries that a strategy would produce, and assessing that the results were unsatisfactory, even without running any user study. Our goal now is to give the reader a glimpse into the kind of summaries generated by each strategy. Hence, in what follows, we will demonstrate those three different strategies by showing how the sentences in Figure 17 are arranged by each strategy. Please keep in mind that only the third strategy was fully developed and used in the evaluation of SongRecommend.

Our first approach to ordering sentences was based on NP-Centrality proposed by Xie (2006) and Xie *et al.* (2008) to represent the importance of a noun phrase in a given text. The value of this measure is obtained by applying *centrality measures* from Social Network analysis (Wasserman and Faust 1994; Corman *et al.* 2002) on an NP network. The construction of the NP network is done both at the sentence level (if two noun phrases are sequentially parsed in a sentence, a link is added between them) and the document level (a link is added to every pair of noun phrases that are parsed out in succession).

According to Xie (2006), NP-Centrality is a correlate of local coherence. Hence, we expected NP-Centrality to help make the summary coherent. We calculated the NP-Centrality of each noun phrase in the review. The centrality of a noun phrase in all the reviews is the sum of its centralities in individual reviews. The sentences are then ordered in descending order of the NP-Centrality of the feature terms they contain. Figure 18 shows the ordering of the sentences from Figure 17 using NP-Centrality. The summary feels incoherent, for example, the description of the instruments is disconnected.

We then moved to a second ordering approach that uses WordNet (Fellbaum 1998), since the WordNet hierarchy has been extensively used to provide and/or to expand lexical items in various Natural Language Processing tasks, including in Natural Language Generation (Bangalore and Rambow 2000; Kleinbauer, Becker and Becker 2007; Mairesse and Walker 2010). Since our features are terms that contain *music or song* as their synonym or hypernym or hyponym in their definitions, we used this information to group the features, as follows:

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

The haunting, middle-eastern tinged "Minarets" sounds like a pre-cursor. The lyrics are pretty cool. "Minarets" is a very dark track. A good saxophone riff sounds a little choppy. "Minarets" has become a rare performance. It has an interesting sound. Dave uses patented hick-up vocals. The song features pulsing drums by Carter. The tone does not fit the rest of the album. The song features the eerie strings.

Fig. 18. Example summary (version 1): ordered via NP-Centrality.

Dave uses patented hick-up vocals.

The haunting, middle-eastern tinged "Minarets" sounds like a pre-cursor. The tone does not fit the rest of the album.

The lyrics are pretty cool.

A good saxophone riff sounds a little choppy. The song features pulsing drums by Carter.The song features the eerie strings.

"Minarets" has become a rare performance. It has an interesting sound. "Minarets" is a very dark track.

Fig. 19. Example summary (version 2): ordered using WordNet.

(1) *Hypernym Group*: The features that contain *music or song* in their hypernyms.
(2) *Synonym Group*: The features that contain *music or song* in their synonyms.
(3) *Hyponym Group*: The features that contain *music or song* in their hyponyms.
(4) *Hypernym Definition Group*: The features that contain *music or song* in their hypernym definition.
(5) *Synonym Definition Group*: The features that contain *music or song* in their synonym definition.
(6) *Hyponym Definition Group*: The features that contain *music or song* in their hyponym definition.

After the f-sentences have been grouped into the above six groups according to the features they contain (clearly not every group will be present for every song), the summary is built by including the groups in the order given above, namely, f-sentences in the Hypernym Group will be included first, then those in the Synonym Group, and so forth. Figure 19 shows the summary as ordered using WordNet, namely, including sentences in the order of the six groups listed above. We see a slight improvement in that the musical instruments are mentioned together. The ordering of the six groups of features also allows us to provide better formatting (the NP-Centrality measure does not lend itself to natural paragraph breaks). However, the summary still lacks in coherence. Hence, we moved to a third ordering approach that exploits a music ontology (Raimond *et al.* 2010). We first extended this music ontology to include few additional concepts that correspond to features identified

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

The haunting, middle-eastern tinged Minaret sounds like a pre-cursor. Minarets is a very dark track.

The lyrics are pretty cool.

"Minarets" has become a rare performance.

It has an interesting sound. The tone does not fit the rest of the album. Dave uses patented hick-up vocals.

The song features the eerie strings. The song features pulsing drums by Carter. A good saxophone riff sounds a little choppy.

Fig. 20. Example summary (final version): ordered via music ontology.

in our corpus such as *chorus*, *vocals*. Also, we extended each of the classes by adding the domain to which it belongs. We identified a total of twenty different domains for all the features and defined a specific order of these domains in order to generate a coherent summary. Table 1 shows the list of domains with examples taken from the ontology – note that not all of these are identified as features by our algorithm. f-sentences are grouped and ordered according to the domain of the features, in the order shown in Table 1. 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, in case feature domains that are considered as sub-domains of another domain appear in the same paragraph, they are ordered next to the features of the parent domain. In Table 1, sub-domains are represented with an *.

Figure 20 shows the final version of our sample summary when the extracted sentences are ordered via this method.

5 Evaluation

As first introduced in Spärck Jones and Galliers (1995) and Mani and Maybury (1999), evaluation of Natural Language systems falls within two types: *intrinsic*, in which their output is evaluated in reference to some ideal model; and *extrinsic*, in which their output is assessed in the context of a specific task a human or machine has to carry out.[5] Each type of evaluation sheds light on different aspects of a Natural Language system, and each type requires different resources. For example, intrinsic evaluation requires gold standards; extrinsic evaluation requires a larger context (e.g., a whole system) in which the components of interest can be evaluated according to the effect they have on a task to be carried out. Extrinsic evaluation is often carried out via user studies. Because of their inherent strengths and weaknesses, we decided to conduct both types of evaluation inasmuch that they are appropriate

[5] This terminology was later refined, e.g., in Spärck Jones (2007), where it was pointed out that these two types should be seen as a continuum rather than as a dychotomy. However, we feel that this typology is useful to categorize how evaluation can be carried out, and we adopt it here.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

Table 1. *Domains of the music ontology*

Order. no.	Domain	Description	Examples
1.	Song	term that is synonymous to the song	track, anthem
2.	Part	different parts of the song	lyrics, lines
3.	Performance	a performance event	live performance
4.	Arrangement	arrangement of the work or event	score
	NotesArrangement*	deals with arrangement of specific notes of the song	melody, tune
5.	Sound	sound in the song	sound
	SoundDetail*	sound-related details	tone, vocals
6.	Time	time-related information	sample rate, instant
7.	Event	event-related information	show
8.	Instrumentation	instruments used	guitar, drums
9.	Personnel	people related	person
	DirectPersonnel*	people directly involved in song creation	musicartist, composer
	External Personnel*	people not directly involved in the song creation	producer, publisher
10.	ExternalWebsite	URL of web sites related to the song	www.myspace.com
11.	Album	one or more songs issued together	album
12.	Storage	related to medium of storage	vinyl, magnetic tape
13.	Transmission	modes of transmission over the network	stream, torrent
14.	Movement	related to movement in a musical work	movement number
15.	Marketing	body related to marketing of the musical work	distributor
16.	Miscellaneous	all others	biography

and feasible. We believe that pooling together results of these different evaluations can give us a better picture of the whole work. Hence, we conducted an intrinsic evaluation of the song title and feature extraction steps; a second intrinsic evaluation of the informativeness of summaries; and, as an extrinsic evaluation, a user study to judge how summaries affect decision-making.

Note that we did not conduct an intrinsic evaluation of each module within the Summary Generation module proper (see Figure 5): sentence partition and regeneration, grouping, sentence selection and ordering. First, these three modules taken together constitute the sentence realization component of SongRecommend. Sentence realization is a task within Natural Language Generation (NLG) that is in

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

general evaluated holistically, and not subcomponent by subcomponent. Second, it is not clear how each of these three modules could be individually evaluated other than by manual inspection by a human judge, since we do not have a reference corpus of partitioned and regenerated sentences. None of the tasks performed by those three modules is a natural task for people, hence a corpus of partitioned and regenerated sentences can only be elicited by assigning subjects to perform this artificial task (as opposed to say the task of writing a review for a song, which is a natural task). Finally, each of the three submodules performs a relatively simple task which does not appear to be evaluated independently.

Hence, we chose to perform an intrinsic evaluation of the generated summaries, and not of their intermediate representations. In addition, our intrinsic evaluation focuses on the informativeness of the generated summaries, rather than their grammaticality and coherence.

More important in our opinion, as concerns the NLG subcomponent, is the extrinsic evaluation of the whole system with users, which demonstrates that the system performs its task sufficiently well. An extrinsic evaluation is well in line with how most of the evaluation is conducted in NLG. In addition, a positive extrinsic evaluation indirectly quells concerns regarding the mistakes that each module in the pipeline unavoidably makes – as concerns the NLG components, these mistakes will result in repetitions, and incoherence of different kinds, from referential to semantic to rhetorical. For example, in Figure 20, the first sentence mentions 'Minaret' (singular) but the second and the fourth refer to 'Minarets' (plural). In the last paragraph, *The song* is used twice, instead of substituting the second occurrence with a pronoun. The fourth sentence juxtaposes *interesting sound* with the comment about the tone not fitting the rest of the album, a possible semantic incoherence. If these errors were to unduly affect the final text, this would be reflected in the evaluation – either in task-oriented measures, or in subjective measures such as user satisfaction. Since our results are promising, we can assume that these mistakes were not too frequent, and when they occurred, they were well tolerated by the users.

5.1 Song title and feature extraction

The song title and feature extraction algorithms (Sections 4.1 and 4.2) were manually evaluated on 100 reviews randomly taken from the corpus (2 or 3 reviews for each album). This relatively small number is due to the need to conduct the evaluation manually. The 100 reviews contained 1,304 occurrences of song titles and 898 occurrences of song features, as previously annotated.

In all 1,294 occurrences of song titles were correctly identified; in addition, 123 spurious occurrences were also identified. This results in a precision of 91.3 percent, and recall of 99.2 percent. The ten occurrences that were not identified contained multiple spelling mistakes in the same word – this case also covers abbreviations such as *Dr*. for Doctor. Recall that we do not clean up mistakes in the reviews, and filter out words that have edit distance *greater* than one from words in titles.

Most of the additionally identified occurrences follow all the criteria of being a song title, as discussed earlier in Section 4.1. However, they are not song titles when

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

Table 2. *Extraction results (100 reviews)*

	Precision	Recall
Song titles	91.3%	99.2%
Features	95.4%	94.9%

the context is taken into consideration. For example, our algorithm recognizes both occurrences of *Are You Experienced?* as song titles in sentences (a) and (b):

(a) *'Are You Experienced?' is an excellent pop album.*
(b) *The lyrics of the track 'Are you Experienced?' are wonderfully written.*

Of the 898 occurrences of song features, 853 were correctly identified by our feature extraction algorithm, with an additional forty-one spurious occurrences. This results in a precision of 95.4 percent and a recall of 94.9 percent. Note that a feature is considered as correctly identified if its head noun is annotated in a review for the song with correct ID. We speculate that both spurious and non-identified occurrences may be due to

(1) TextTiling: As discussed in Section 4.2, we use the discourse units returned by TextTiling to indirectly identify if a feature term is attributed to a song and do not perform actual reference resolution.
(2) WordNet: Since feature terms are extracted with the help of WordNet, some terms are wrongly identified as features, and some terms are not identified.

Our extraction results for both song titles and features are summarized in Table 2.

As a baseline comparison, we implemented the feature extraction algorithm from Hu and Liu (2004).[6] We compared their algorithm with ours on ten randomly chosen reviews from our corpus, for a total of about 500 sentences. Its accuracy (40.8 percent precision, and 64.5 percent recall) is much lower than ours. For a fairer comparison, we incorporated TextTiling and our heuristics into the Hu and Liu algorithm (2004), increasing its precision to 56.04 percent but with no effect on recall. On this subset of the data, our algorithm still performs much better (93.2 percent precision and 91.5 percent recall) – see Table 3. In addition, the performance of the algorithm from Hu and Liu (2004) on our data is much lower than their original results on product reviews (72 percent precision, and 80 percent recall). We believe that this is due to the different nature of song reviews with respect to product reviews, as we discussed in Section 2.

5.2 Informativeness of the summaries

Having evaluated the extraction components of our work, we set out to evaluate the summaries in their entirety. In the last fifteen years, summary evaluation has been a

[6] We implemented the Hu and Liu algorithm (2004) ourselves since the original code was not publicly available.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

Table 3. *Hu and Liu (2004) versus SongRecommend (feature extraction, ten reviews)*

Method	Precision	Recall
Hu and Liu (2004)	40.8%	64.5%
Hu and Liu (2004) + TextTiling and heuristics	56.04%	64.5%
SongRecommend	93.2%	91.5%

hot topic of research (Mani and Maybury 1999). Intrinsic evaluation of summaries generally concerns *quality* (how the summary reads) and *informativeness*. Quality is mostly evaluated by asking subjects to score the summaries (our user study will touch on judging the quality of the summaries; see Section 5.3). Informativeness can either concern how much the source information is preserved at various compression ratios or how much the summary is similar to an ideal output. For the latter kind of evaluation, a completely automatic method, called ROUGE (Lin 2004), uses n-gram recall between a candidate summary and a set of reference summaries. In contrast, the Pyramid method (Nenkova and Passonneau 2004; Harnly *et al.* 2005) focuses on semantic coverage, but calls for manual annotation of *Summary Content Units* (SCUs) in reference summaries. Those SCUs are arranged in a pyramid of tiers of importance. The informativeness of a generated summary is then evaluated in terms of its coverage of SCUs, which are partially ordered according to their importance, and in relation to the length of the summary.

ROUGE and Pyramid sit at the two ends of a spectrum as concerns evaluation measures: ROUGE is rather coarse, since it is based on n-gram recall, whereas Pyramid is much more semantic oriented; however, ROUGE does not require further extensive manual annotation as Pyramid does. Both measures, though, rest on the availability of a corpus of reference summaries, which does not exist for our task. Whereas manual creation of corpora of reference summaries has been undertaken, it generally happens with strong institutional support: e.g., the DUC competitions in the United States are backed by the Advanced Research and Development Activity (ARDA), and are run by the National Institute of Standards and Technology (NIST). Note that in our case, we would need to generate at least one reference summary *per song* in our twenty-seven albums, for a total of at least 391 summaries (our twenty-seven albums contain 391 songs altogether, see Section 4).

Hence, we decided to evaluate the informativeness of our summaries in an indirect way, by comparing the information they provide to that included in high-quality reviews as those written by professional critics. We chose two or three critic reviews per album taken from www.pitchfork.com, www.rollingstone.com, and www.allmusic.com. Please note, similar to the reviews in our corpus, these critic reviews discuss the *whole* album, hence we do not have high-quality reviews for each individual song: if they existed, we could use critic reviews of each individual song as the gold standard to compare our summaries against. However, given the nature of the critic reviews, we assume they cover important features of the songs in those albums, and hence we use them for our indirect evaluation of informativenes.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

We randomly selected ten albums, and from each album we selected five songs, and generated the corresponding fifty summaries. Summaries for two additional songs were added, since they had already been generated for other reasons. This resulted in a test set of fifty-two summaries, which corresponds to summarizing about 600 album reviews, since to create each of these fifty-two summaries, fifty reviews are summarized (recall that each album has fifty reviews, see Section 4).

We manually annotated both critic reviews and the automatically generated summaries for song titles and song features. Altogether 302, i.e., 91.2 percent 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 noun phrase is identified by both the critic review and the summary, and *attributed to the same song*). Sixty-four additional features were included in the generated summaries, for a recall of 82 percent. It is not surprising that additional features may appear in the summaries: even if only one of the fifty 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 addition, we believe that in this case precision should be more important than recall, since the gold standard texts we are using are not summaries tailored to specific songs.

5.3 Between-Subjects user study

Our intrinsic evaluation gives satisfactory results. However, we believe that 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-subjects 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 summaries, as shown earlier in the paper in Figure 4. 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 fifty reviews for that album whose length is closest to the average length of album reviews in the corpus (478 words; see Figure 6).

Each user was presented with five songs in succession, with three recommendations each (only the top three recommendations were presented among the available twenty-five; 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 were open and closed etc. After using the system, users were administered a brief questionnaire, which included questions on a 5-point Likert scale. Eighteen users interacted with the baseline version and twenty-one with the experimental version. All users were students at our university. About 85 percent of them were graduate students, about 3/4 were males, and about 2/3 listened to music daily.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

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 took less time to make their decision when compared with baseline subjects ($p = 0.019$, $t = 2.510$). We also examined differences in time-on-task due to gender, previous knowledge of music, or education level (via t-test or one way ANOVA). We found that none of these factors affected time-on-task (this lack of dependence should be considered as tentative, since some of the groups, e.g., undergraduate students, are very small).

We consider the decrease in time to selection as a positive result, given that music collections can include millions of songs. However, time-on-task represents the time it takes users to peruse the review or summary, which in turn is affected by the length of the review or summary: 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 one, and the difference between the two conditions is significant ($\chi^2 = 8.74$, degrees of freedom(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. In particular, negative opinions are much more evident in the review summaries.

For example, Figure 21 shows the song 'Sunday Bloody Sunday' from the album *War* by U2 (first row in the figure) and the recommendations presented to the user (rows 2 through 4): 'Surrender' from the same album, and 'Pride' and 'Bad', both from *Unforgettable Fire*, another album by U2. On the right, we show the summaries generated by SongRecommend. On the left, we include the relevant portions of the corresponding album reviews, and indicate how many words occur in the album review before the portion on the song in question, and how many occur afterwards (note that users were presented with the whole album review, not only with the excerpt that we highlight here). As the reader may notice, the same album review is shown for both 'Pride' and 'Bad'; this derives from our choice of showing the album review closest to the average length of the album reviews in our corpus. In this specific case, since both 'Bad' and 'Pride' are mentioned in the same sentence, the specific excerpts that we highlight are the same as well.

The middle column in Figure 21 (rows 2–4) shows how many users in the baseline and experimental conditions chose that specific recommendation. Thirteen users in the baseline condition picked the first recommendation ('Surrender'), while only three users picked it in the experimental version. These numbers are reversed for the second recommendation ('Pride', chosen by four users in baseline condition but by twelve in experimental condition); and for the third recommendation ('Bad', six

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

Table 4. *Length of summary versus number of selections*

Length	Selected	Not selected
Longest	49	35
Medium	39	45
Shortest	31	53

users in baseline condition but thirteen in experimental condition). This example also shows that whereas users need to wade through a potentially high number of words in the album reviews before getting to the song of interest, the number of words in the summaries is much lower (31, 7, 41, 36 words respectively). Also, for each recommended song, only a few lines talk about the actual song and those lines may be buried deep inside the album review. On the other hand, the summaries make it apparent which features are shared between the original song ('Sunday Bloody Sunday') and the recommended songs: e.g., the political theme and the word *anthem* (for 'Pride'), and *Larry Mullen* (for 'Bad').

Since the summaries are so much shorter than the original reviews, and length affected time-on-task in the between-subject comparison, a natural question arises: Does the relative length of different summaries affect users' decisions? In other words, could users simply choose the song associated with the shortest summary? The answer is no.

We categorized three recommendations for each song as longest, medium, and shortest based on the length of the review summary. Table 4 shows how many times a summary of relative length was chosen – recall that there are twenty-one users and four songs for which to choose one or more recommendations. The table shows that users chose the shortest summary less frequently, and a χ^2 test confirms that this difference is significant ($p = 0.02$, $\chi^2 = 7.77$, $df = 2$).[7] This indicates that users do not merely follow a recommendation because it has the shortest summary.

Usability measures. As mentioned, after using the system, users were administered a brief questionnaire which included questions on a 5-point Likert scale. The questionnaires included three common questions between the two conditions, which differed only as concerns the item being asked about:

(1) Did you feel that the *album review/summary* helped you quickly decide if you want to listen to that song?
(2) Did you feel that the *album review/summary* is of decent length?
(3) Did you feel that the *album review/summary* has too much information about the song?

The experimental subjects gave a more positive assessment of the length of the summary than the baseline subjects of the length of the album review (question (2),

[7] The events in the table are not strictly independent, since each user is presented with four songs, and for each, can choose more than one recommendation.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

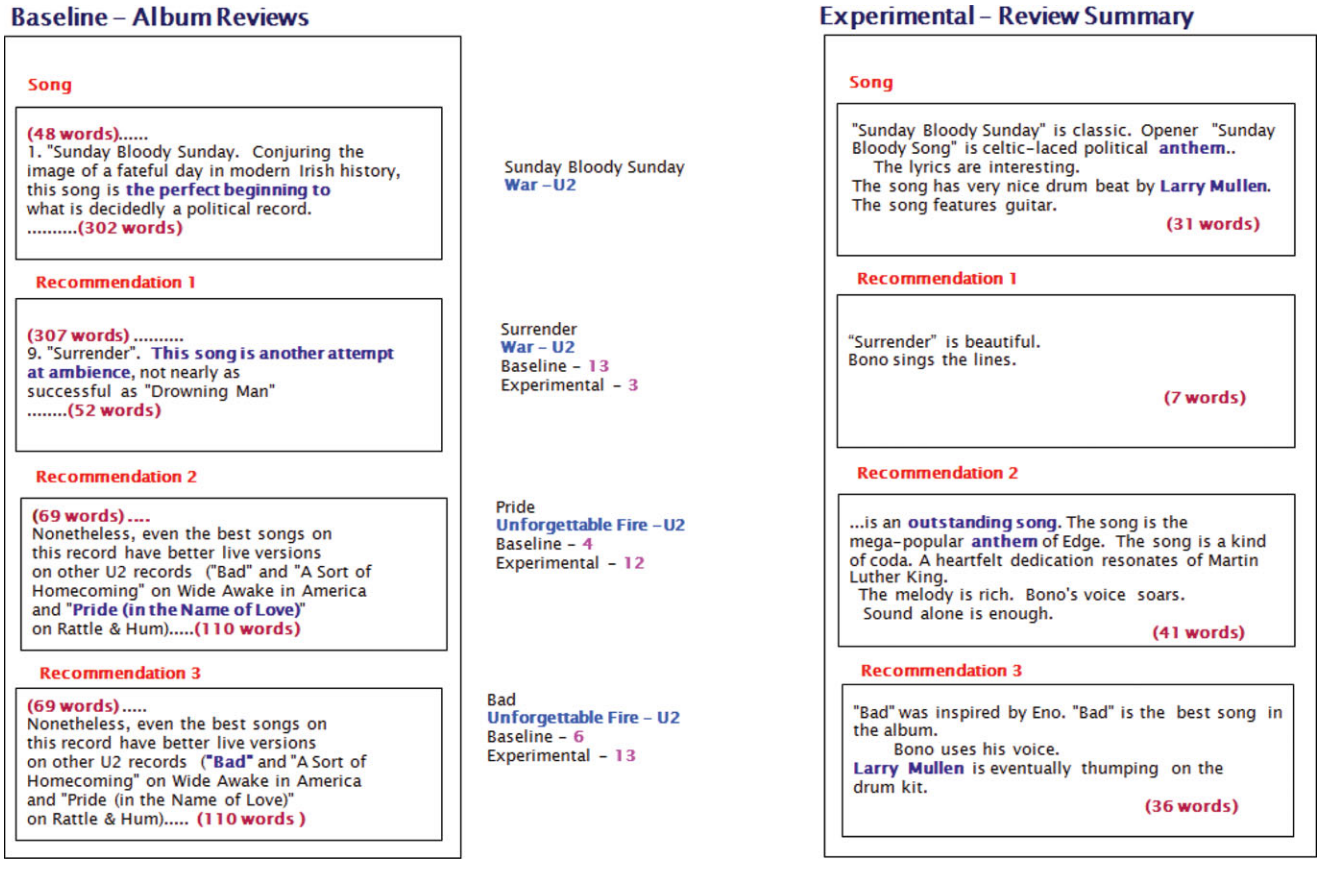


Fig. 21. (Colour online) Users' choices between album reviews (left), and review summaries (right).

$p = 0.003$, $t = -3.248$, $df = 31.928$). There were no significant differences on the other two questions.

A multiple Linear Regression with time-on-task as dependent variable and as predictors, the number of words in the album review/review summary, and the three questions they all answered, revealed only one, not surprising, correlation: the number of words the user read correlates with time-on-task ($R^2 = 0.277$, $\beta = 0.509$, $p = 0.004$). This correlation is not surprising, since we had already established that there is a significant difference between the number of words in the original reviews and in our summaries.

Users in the experimental version were also asked to rate the grammaticality and coherence of the summary (together with the question on length, these assessments can be considered as an intrinsic evaluation of the quality of the summary, as mentioned earlier). The two questions users were posed are: 'Did you feel that the summary is grammatical?' and 'Did you feel that the overall flow of the summary is good?' 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.

Finally, users could add open comments. Whereas the numbers are too small to draw other than tentative conclusions, we note that twelve of the baseline users commented that they would have liked the Music RS better if they had been presented with information specific to the song; and ten of the experimental users commented that the song review summary was helpful to their decision-making process.

6 Conclusions and future work

In this paper, we have described SongRecommend, a system that uses information extraction and summarization techniques to generate summaries of individual songs from multiple album reviews. With respect to Product RSs, our Music RS addresses challenges that arise both from the features of songs seen as 'products' and from the fact that in general song reviews are not available *per se*, but are scattered within the album reviews to which a specific song belongs. In addition, most work on product recommendation has been conducted from a text mining point of view, and little attention has been paid to summarizing those reviews in a coherent summary. Whereas the techniques we have used are state-of-the-art, the contribution of our work is integrating them into an effective end-to-end system. We presented several independent evaluations of our work. First, we evaluated it intrinsically as concerns information extraction, and the informativeness of the summaries. In all cases, we showed that most of our precision and recall results are above 90 percent. We also ran an extrinsic evaluation via a user study. Users made quicker decisions and their choice of recommendations was more varied when presented with song review summaries than with album reviews. Whereas the smaller size of the summary has an impact on time-on-task, users do not appear to choose a specific recommendation based only on number of words. Our surveys, although informally, suggest that

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

users in the experimental condition found the summaries acceptable, as concerns length, grammaticality, and coherence. Experimental users also indicated that the summaries helped them choose the songs they were most interested in.

As concerns future work, to become a truly usable tool, SongRecommend needs to include a more sophisticated recommendation model. We mentioned earlier that a hybrid of content- and context-based filtering would be most appropriate. SongRecommend monitors songs listened to by the user on several portable and non-portable music devices and recommends songs which may either exist in the user's collection or from the Internet; at the same time, the review summaries that SongRecommend generates could be used to rank recommendations.

Additionally, the recommendation model should take user preferences into account. As a matter of fact, an additional module in our architecture uses the Apriori Algorithm (Mitchell 1997), an association mining algorithm, to learn preference patterns among the attributes of the songs listened by the user. Those patterns were then used in one version of SongRecommend to re-rank recommendations by similarity between the song and frequent patterns in the preferences – see Tata (2010) for further details. In spite of the existence of this module, user preferences were not included in our experiments, because the frequent patterns we had been able to mine were based on limited data, and only concerned whether the user would keep listening to songs in the same album. However, it is a natural next step to build on the foundations that we have already set up in our system for mining user preferences.

A second venue of the future research is improving the NLG components. For example, as pointed out earlier, our NLG module does not include referring expression generation, a task that has received considerable attention in the NLG literature (Dale and Reiter 1995; Krahmer, Erk and Verleg 2003; Belz *et al.* 2010). The generation of referring expressions would of course be affected by grouping and ordering, since the proper NP form to choose (pronoun, definite description, proper noun) also depends on the distance between a referent and its subsequent mentions. At the same time, referring expression generation would affect sentence content, since more appropriate referring expressions would substitute (some of) the noun phrases originally contained in the sentence in question. Our modular approach, the availability of the Typed Dependency parse for the original sentence, and the usage of YAG all facilitate in being able to slice out the original noun phrases and substitute them with more appropriate expressions.

Beyond referring expressions, the coherence of our summaries could be improved by signalling rhetorical relations that emerge from the juxtaposition of sentences, including contrasting opinions, and the logical organization of features. Some of these rhetorical relations can be signalled via appropriate cue words. Whereas full content planning is beyond the scope of this work, some mileage could be obtained by adapting the discourse parsing work that our group has engaged in in the past (Di Eugenio, Moore and Paolucci 1997; Subba 2007; Subba and Di Eugenio 2009).

In the longer term, we intend to apply our methodology to different domains. We believe that our framework is applicable to any domain where reviews of individual components need to be summarized from reviews of collections, such

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

as travel reviews that cover many cities in a country, or different restaurants in a city. Whereas the extraction component may not seem appropriate for sites, such as TripAdvisor, that aggregate reviews and present them associated to individual items (hotels, restaurants), it would be useful for narrative travel guides such as those published by the *New York Times* (www.nytimes.com/pages/travel/index.html) or for travel forums such as the one at www.virtualtourist.com.

References

Bangalore, S., and Rambow, O. 2000. Corpus-based lexical choice in natural language generation. In *Proceedings of ACL 2000, the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, pp. 464–71.

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

Belz, A., Kow, E., Viethen, J., and Gatt, A. 2010. Generating referring expressions in context: the GREC shared task evaluation challenges. In E. Krahmer and M. Theune (eds.), *Empirical Methods in Natural Language Generation*, pp. 294–327. Lecture Notes in Computer Science, Vol. 5980. Berlin, Germany: Springer.

Benamara, F., Cesarano, C., Picariello, A., Reforgiato, D., and Subrahmanian, V. 2007. Sentiment analysis: adjectives and adverbs are better than adjectives alone. *Proceedings of the International Conference on Weblogs and Social Media* (ICWSM), Boulder, CO, USA.

Bruce, R., and Wiebe, J. 1999. Recognizing subjectivity: a case study of manual tagging. *Natural Language Engineering* **5**(2): 187–205.

Cano, P., Koppenberger, M., and Wack, N. 2005. An industrial-strength content-based music recommendation system. In *Proceedings of the 28th Annual International ACM SIGIR Conference* (SIGIR 2005), Salvador, Brazil, pp. 673–73.

Carenini, G., Ng, R., and Pauls, A. 2006. Multi-document summarization of evaluative text. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics* (EACL 2006), Trento, Italy, pp. 305–12.

Carletta, J. 1996. Assessing agreement on classification tasks: the Kappa statistic. *Computational Linguistics* **22**(2): 249–54.

Celma, Ò. 2006. *Interaction Design for Recommender Systems*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain.

Celma, Ò. 2010. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Berlin, Germany: Springer.

Corman, S., Kuhn, T., McPhee, R., and Dooley, K. 2002. Studying complex discursive systems: centering resonance analysis of organizational communication. *Human Communication Research* **28**(2): 157–206.

Dale, R., and Reiter, E. 1995. Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cognitive Science* **18**: 233–63.

de Marneffe, M.-C., and Manning, C. D. 2008. Stanford typed dependencies manual. http://nlp.stanford.edu/software/dependencies_manual.pdf. (Accessed 16 Sep 2012).

Di Eugenio, B., Moore, J. D., and Paolucci, M. 1997. Learning features that predict cue usage. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (ACL-EACL97), Madrid, Spain, pp. 80–7.

Ding, X., Liu, B., and Yu, P. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the International Conference on Web Search and Web Data Mining*, Palo Alto, CA, USA, pp. 231–40.

Downie, J. S., and Hu, X. 2006. Review mining for music digital libraries: phase II. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, Chapel Hill, NC, USA, pp. 196–97.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

Esuli, A., and Sebastiani, F. 2006. SentiWordNet: a publicly available lexical resource for opinion mining. *Proceedings of the 5th Conference on Language Resources and Evaluation* (LREC-06), Genova, Italy.

Fellbaum, C. (ed.) 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press.

Gamon, M., Aue, A., Corston-Oliver, S., and Ringger, E. 2005. Pulse: mining customer opinions from free text. In *Advances in Intelligent Data Analysis VI*, Lecture Notes in Computer Science, Vol. 3646, pp. 121–32. Berlin, Germany: Springer.

Goel, S., Broder, A., Gabrilovich, E., and Pang, B. 2010. Anatomy of the long tail: ordinary people with extraordinary tastes. In *Proceedings of the Third International Conference on Web Search and Web Data Mining*, New York, NY, USA, pp. 201–10.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. 1992. Collaborative filtering to weave an information tapestry. *Communications of the ACM* **35**(12): 61–70.

Harnly, A., Nenkova, A., Passonneau, R., and Rambow, O. 2005. Automation of summary evaluation by the Pyramid method. *Proceedings of the International Conference on Recent Advances in Natural Language Processing* (RANLP-2005), Borovets, Bulgaria.

Hatzivassiloglou, V., and McKeown, K. R. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (ACL-EACL97), Madrid, Spain, pp. 174–81.

Hearst, M. A. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics* (ACL99), Las Cruces, NM, USA, pp. 9–16.

Higashinaka, R., Prasad, R., and Walker, M. 2006. Learning to generate naturalistic utterances using reviews in spoken dialogue systems. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* (COLING/ACL 2006), Sidney, Australia, pp. 265–72.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD 2004), Seattle, WA, USA, pp. 168–77.

Jin, W., Ho, H., and Srihari, R. 2009. OpinionMiner: a novel machine learning system for web opinion mining and extraction. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD 2009), Paris, France, pp. 1195–204.

Klein, D., and Manning, C. D. 2002. Fast exact inference with a factored model for natural language parsing. In *Proceedings of Advances in Neural Information Processing Systems 15* (NIPS 2002), Vancouver, Canada, pp. 3–10.

Kleinbauer, T., Becker, S., and Becker, T. 2007. Combining multiple information layers for the automatic generation of indicative meeting abstracts. In *Proceedings of the Eleventh European Workshop on Natural Language Generation* (ENLG 07), Schloss Dagstuhl, Germany, pp. 151–54.

Krahmer, E., Erk, S., and Verleg, A. 2003. Graph-based generation of referring expressions. *Computational Linguistics* **29**(1): 53–72.

Le Roux, F., Elkunchwar, R., Ghai, V., Gao, Y., and Lu, J. 2007. A course recommender system using multiple criteria decision making method. *Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering* (ISKE 2007), Chengdu, China.

Levy, M., and Bosteels, K. 2010. Music recommendation and the long tail. *Proceedings of the First Workshop on Music Recommendation and Discovery* (at ACM RecSys) (WOMRAD 2010), Barcelona, Spain.

Lin, C.-Y. 2004. ROUGE: a package for automatic evaluation of summaries. In S. S. Marie-Francine Moens (ed.), *Proceedings of the Workshop Text Summarization Branches Out (at ACL 2004)*, Barcelona, Spain, pp. 74–81.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

Mairesse, F., and Walker, M. 2010. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction* **20**(3): 227–78.

Mani, I., and Maybury, M. T. 1999. *Automatic Summarization*. Boston, MA, USA: The MIT Press.

McRoy, S., Channarukul, S., and Ali, S. 2003. An augmented template-based approach to text realization. *Natural Language Engineering* **9**(4): 381–420.

Miller, B., Albert, I., Lam, S., Konstan, J., and Riedl, J. 2003. MovieLens unplugged: experiences with an occasionally connected recommender system. In *Proceedings of the 8th International Conference on Intelligent User Interfaces* (IUI 2003), Miami, FL, USA, pp. 263–66.

Miller, G. A., Chodorow, M., Landes, S., Leacock, C., and Thomas, R. G. 1994. Using a semantic concordance for sense identification. In *Proceedings of the Workshop on Human Language Technology* (HLT '94), Plainsboro, NJ, USA, pp. 240–43.

Minnen, G., Carroll, J., and Pearce, D. 2000. Robust, applied morphological generation. In *Proceedings of the 1st International Natural Language Generation Conference* (INLG 2000), Mitzpe Ramon, Israel, pp. 201–8.

Mitchell, T. 1997. *Machine Learning*. Burr Ridge, NJ, USA: McGraw Hill.

Nastase, V. 2008. Topic-driven multi-document summarization with encyclopedic knowledge and spreading activation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP 2008), Edinburgh, Scotland, pp. 763–72.

Nenkova, A., and Passonneau, R. 2004 . Evaluating content selection in summarization: the Pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (HLT-NAACL 2004), Boston, MA, USA, pp. 145–52.

Nguyen, P., Mahajan, M., and Zweig, G. 2007. Summarization of multiple user reviews in the restaurant domain. Technical Report MSR-TR-2007-126. Microsoft, Redmond, WA, USA.

Popescu, A., and Etzioni, O. 2005. Extracting product features and opinions from reviews. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing* (HLT/EMNLP 2005), Vancouver, BC, Canada, pp. 339–46.

Raimond, Y., Giasson, F., Jacobson, K., Fazekas, G., Gängler, T., and Reinhardt, S. 2010. Music ontology specification. Specification document. http://musicontology.com/. (Accessed 16 Sep 2012).

Ramshaw, L., and Marcus, M. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, Cambridge, MA, USA, pp. 82–94.

Rich, E. 1979. User modeling via stereotypes. *Cognitive Science* **3**(4): 329–54.

Saggion, H. 2011. Learning predicate insertion rules for document abstracting. In A. Gelbukh (ed.), *Computational Linguistics and Intelligent Text Processing*, pp. 301–12. Lecture Notes in Computer Science, Vol. 6609. Berlin, Germany: Springer.

Saggion, H., and Funk, A. 2010. Interpreting SentiWordNet for opinion classification. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias (eds.), *Proceedings of the Seventh International Conference on Language Resources and Evaluation* (LREC'10), Valletta, Malta.

Saggion, H., and Lapalme, G. 2002. Generating indicative-informative summaries with SumUM. *Computational Linguistics* **28**(4): 497–526.

Schedl, M., Widmer, G., Pohle, T., and Seyerlehner, K. 2007. Web-based detection of music band members and line-up. *Proceedings of the 8th International Conference on Music Information Retrieval* (ISMIR 2007), Vienna, Austria.

Soubbotin, M., and Soubbotin, S. 2005. Trade-off between factors influencing quality of the summary. *Proceedings of the Document Understanding Workshop* (DUC 2005), Vancouver, BC, Canada.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press

Spärck Jones, K. 2007. Automatic summarising: the state of the art. *Information Processing and Management* **43**(6): 1449–81.

Spärck Jones, K., and Galliers, J. R. 1995. *Evaluating Natural Language Processing Systems: An Analysis and Review*. Lecture Notes in Computer Science, Vol. 1083. New York, USA: Springer.

Subba, R. 2007. Exploiting event semantics to parse the rhetorical structure of natural language text. In *Proceedings of the Doctoral Consortium at NAACL-HLT 2007, the Conference of the North American Chapter for the Association for Computational Linguistics*, Rochester, NY, USA, pp. 21–4.

Subba, R., and Di Eugenio, B. 2009. An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, CO, USA, pp. 566–74.

Tata, S. 2010. *SongRecommend: A Music Recommendation System with Fine-Grained Song Reviews*. PhD thesis, University of Illinois at Chicago, IL, USA.

Tintarev, N., and Masthoff, J. 2007. Effective explanations of recommendations: user-centered design. In *Proceedings of the ACM Conference on Recommender Systems* (RecSys'07), Minneapolis, MN, USA, pp. 153–56.

UPN 2008. UPnP Device Architecture Version 1.0. www.upnp.org. (Accessed 16 Sep 2012).

Van Meteren, R., and Van Someren, M. 2000. Using content-based filtering for recommendation. *Proceedings of the ECML/MLNet Workshop on Machine Learning and the New Information Age*, Barcelona, Spain.

Van Setten, M., Pokraev, S., and Koolwaaij, J. 2004. Context-aware recommendations in the mobile tourist application COMPASS. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 515–48. Lecture Notes in Computer Science, Vol. 3137. Berlin, Germany: Springer.

Wasserman, S., and Faust, K. 1994. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences Series, no. 8. Cambridge, UK: Cambridge University Press.

Wiebe, J. M., Bruce, R. F., and O'Hara, T. P. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (ACL99), College Park, MD, USA, pp. 246–53.

Xie, Z. 2006. *Machine Learning in Automatic Text Summarization: From Extracting to Abstracting*. PhD thesis, University of Illinois, Chicago, IL, USA.

Xie, Z., Di Eugenio, B., and Nelson, P. C. 2008. From extracting to abstracting: generating quasi-abstractive summaries. In *Proceedings of the Sixth International Language Resources and Evaluation* (LREC'08), Marrakech, Morocco.

Zhuang, L., Jing, F., Zhu, X., and Zhang, L. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management* (CIKM 2006), Arlington, VA, USA, pp. 43–50.

https://doi.org/10.1017/S1351324912000253 Published online by Cambridge University Press