Audio Metaphor: Audio Information Retrieval for Soundscape Composition

Miles Thorogood
SIAT, Simon Fraser University
250 -13450 102 Avenue
Surrey B.C., Canada
mthorogo@sfu.ca

Philippe Pasquier
SIAT, Simon Fraser University
250 -13450 102 Avenue
Surrey B.C., Canada
pasquier@sfu.ca

Arne Eigenfeldt
SIAT, Simon Fraser University
250 -13450 102 Avenue
Surrey B.C., Canada
eigenfel@sfu.ca

ABSTRACT

We have developed an audio information retrieval system, named Audio Metaphor. It is designed for use in soundscape composition. Audio Metaphor accepts natural language queries, and recommends audio files from online collaborative databases. Audio Metaphor uses a sub-query generation algorithm, named SLiCE (String List Chopping Experiment) that accepts a word-feature list, parsed from a natural language query by another component of the Audio Metaphor system. A set of audio file recommendations is returned to the soundscape composer.

Audio Metaphor has been used to recommend recordings from a database of audio files by using the descriptions associated with each recording. It has also been applied in a live context within a contemporary dance production *In[a]moment*, to provide audio files for soundscape composition. Audio Metaphor facilitated audience interaction by listening for Tweets that the audience addressed to the performance; in this case, it processed the Twitter feed in realtime to recommend audio files to the soundscape composer.

1. INTRODUCTION

Current music technology can facilitate the use of sound recordings for the creation of rich sound compositions. Programming environments such as SuperCollider [1], PD [2], and MaxMSP [3] provide a large number of programming interfaces to make the artistic combination and processing of audio recordings accessible.

At the same time, there is an exponentially growing amount of data available online, created by individuals sharing information in online collaborative environments, such as Freesound [4] and Twitter [5]. Much of that data could be used in soundscape composition. Soundscape composition is the artistic combination and processing of soundscape recordings to impress upon the listener a real or imagined soundscape. Soundscape is the term used to describe the sounds present within a locale at a given time. A large number of the recordings in Freesound are tagged as soundscape (4482), or as field recordings (14329) and so could be used in soundscape compositions.

Other online data sources could be used to construct themes for soundscape compositions; for example trending Tweets could be mined to provide possible concepts for the creation of a new composition.

This rich source of data and inspiration is currently underutilized in soundscape composition because there is a lack of tools for extracting useful subsets of the data and interfacing with composition systems. We have developed a system to search online databases to recommend sources for use in soundscape composition. This system, named Audio Metaphor, accepts and processes natural language queries, which are used to generate database queries that return audio files semantically related to the initial query.

The words in the natural language query are processed to generate audio-file recommendations; however, these recommendations may be narrow in scope or number. To broaden the scope and number of results, Audio Metaphor generates a second set of database queries by searching Twitter for Tweets using the words in the natural language query. This use of Tweets adds associated words to the search, and as a result, a larger conceptual space is explored by the algorithm. This exploration returns a greater number of recommendations across a wider space, facilitating richer creative decisions for the soundscape composer.

In Section 2, we provide further context for the development of Audio Metaphor by reviewing related works. Section 3 discusses the underlining algorithms used to develop Audio Metaphor. Section 4 describes how Audio Metaphor was employed in the contemporary dance and technology production *In[a]moment* as a performance based application system. In section 5, we evaluate the effectiveness of the system based on its use in this performance.

2. RELATED WORK

To date, audio information retrieval for soundscape recordings has only been approached by a few researchers, while utilizing online collaborative and social media databases is an unexplored area. Eigenfeldt and Pasquier [6] demonstrate how audio information retrieval can be effectively used for soundscape composition. Composition by their system is negotiated by performance agents that choose soundscape recordings from a database of 227 hand selected files. Composition decisions are based upon audio analysis features, and a small set of tags entered by the researchers. Our system differs from this research by using an online database with unstructured user contributed tags, as well as generating queries from social media posts.

Using unstructured databases for soundscape creation is demonstrated by Finney and Janer [7], who describe an autonomous soundscape system for virtual reality environments that use the Freesound database to augment scenes from Google Street View with sounds. Words that represent objects in the scene, referring to sound sources, are used to search the database for recordings, which are then manipulated into a total soundscape for that scene. Resulting soundscapes were found to represent the scene effectively for subjects when evaluated; however, the system is strongly administered by the researchers and is run offline. Audio Metaphor differs by the processing and representation of natural language queries, and using social media to generate related queries, where the results are given to a performer in a realtime context.

Social media users have produced a vast amount of data with a diverse range of topics. Twitter, a micro-blogging service, provides an accessible platform for millions of discussions and sharing of experiences in the form of news or conversations [8] through short text-based posts. Previous work has shown that microblogging entries have a range of qualities and authority that develop through the social structure of those networks [9–11]. Our system does not pay attention to the quality or authority of single posts; instead we take the most recent posts related to a query and extract the major text features to generate additional queries for soundscape recording retrieval.

Content sharing services, such as YouTube, Flickr, and Freesound, facilitate the exchange of knowledge through a variety of forms of media. Several audio database projects have arisen in recent years that leverage online users proclivity for generating content [4, 12–14]. Users will also add metadata to include tags and descriptions, of the content. Tags and descriptions entered by contributors have been used by Liqiang [15] to enrich text-based online community dialogue with multimedia content. The conceptual information that may be inferred from tags and descriptions was exploited by Wang [16] in the mood classification of music. Also with the domain of mood classification from user tags, Stockholm and Pasquier [17] develop a system for reinforced learning to categorize audio files during a performance.

Audio Metaphor uses unstructured databases and social media. Audio Metaphor first extracts word-features from natural language queries and searches for audio recordings in unstructured databases. Second, word-features are used to generate additional queries by searching social media posts to explore a larger semantic space around the natural language query.

3. SYSTEM DESCRIPTION

Audio Metaphor systematically carries out several steps to process natural language queries and present back to the user audio file recommendations. First, a word-feature list is created from the natural language query. These word-features are then used as input to the SLiCE algorithm, which uses them to generate database queries. SLiCE then builds sets of recommendations for mutually exclusive queries using sets of word-features in decreasing length.

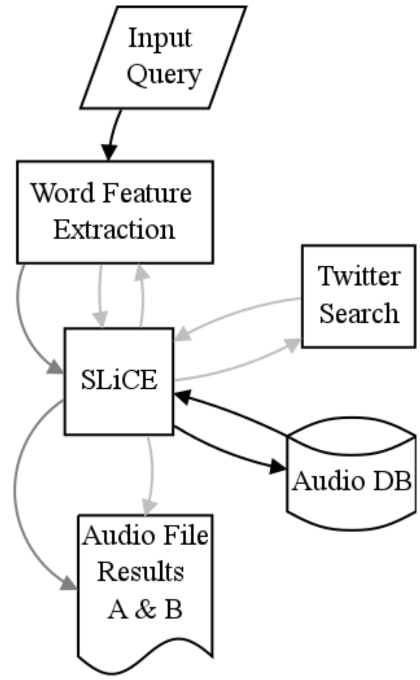

Figure 1. System diagram, showing where a query is input by the user. Word features are are extracted, and on one path, first used to retrieve associated Tweets, then both paths are fed into the SLiCE algorithm which generates sub-queries to search for audio file recommendations. Recommendations from both paths are then presented to the composer for mixing.

The subquery generation algorithms are presented in Section 3.3.

Audio Metaphor uses the same word-features to query Twitter for Tweets containing associated words. Word-features in the Tweets are extracted based upon word frequency, and processed to create an additional word-feature list. This extraction method is described in Section 3.4. The generated list is used to query the audio database in the same way as the original word-feature list. Both set of recommendations are returned by the system and available for use by the composer.

3.1 Audio File Dataset

Audio Metaphor uses Freesound as the database for audio files, and the Freesound API to access that dataset. Freesound is an online collaborative audio file database, aimed at providing a platform for a wide range of users to share audio clips [4]. There are over 120 000 uploaded audio clips that vary in their content (e.g. synthesized tones, field recordings), and quality of entry (accuracy of commentary, recording quality, etc.).

Content presented in the audio file is implied through user contributed commentary and tags, that provide semantic inferences of the content in the media. Although there is no explicit user rating of audio files, a counter for how many times the file has been downloaded is kept - implying the popularity of files - and search results are presented by descending popularity count.

User-entered tags and commentary are indexed by the Freesound search engine. An online web interface allows users to browse the database by searching for keywords present in tags and commentary. The Freesound search engine applies the boolean operator AND to words in a search query that often fails to return a nonempty result for verbose queries. With the Freesound search engine, if all words from the search query are not present in an indexed entry, then an empty search result is returned. Therefore, entering a natural language query often proves overly verbose for the search characteristics of the Freesound search engine.

The Freesound API [18] facilitates using Freesound to access the aforementioned commentary and tags and, additionally, a range of audio analysis features for audio files. Audio Metaphor leverages user entered tags and commentary in searching for an audio representation of natural language queries.

3.2 Word-feature Extraction

Audio Metaphor extracts word features from a natural language query entered in English. A further implementation of a translation tool could be added in the Audio Metaphor process chain to parse other languages. The maximum query input length is 1000 characters, and it was observed through casual testing that users would typically enter descriptions well below this limit without direction. An example description entered by a user could contain several elements:

> ”On hot summer days, down by the river, we would listen to the hum of insects.”

Word-features of the description are considered as nouns and verbs relating to sound events, and adjectives as the modifiers of those events. We choose these word-features because of the signifying properties of sounds associated with words of these types. As such, using these words to search audio files results in recommendations that have the greatest potential for soundscape composition. The above description relates the concepts of a summer day, a river, and a hum of insects. Each concept has a potential sonic representation, which when summed embodies the qualities in the query. By taking this general interpretation of concepts in the query, and omitting subjective associations, a soundscape can be described by the sounds of its subjects, activities, places or things. Thus, important words of the example description would be:

> hot summer days river hum insects

We use the Natural Language Toolkit [19] and employ the WordNet module to extract word-features. Furthermore, the 100 commonest English words found in writing, based upon the Oxford English Dictionary Corpus [20], are removed. The order of occurrence of words in the natural language query is kept, to capture relationships that exist between words that are closer together. Word-features are then input to the SLiCE algorithm to generate sub-lists of word-features for searching a database.

3.3 SLiCE: String List Chopping Experiments

SLiCE generates sub-lists for search queries. These search queries are then used to create mutually exclusive sets of recommendations from a database. SLiCE creates these sets for all the word-features, keeping the word-features grouped together were possible. Successful search results are non-empty sets of recommendations.

An initial query, which is the word-feature list as presented in Section 3.2, is first used to search the database for results. In the event that the number of results returned is less than the user set maximum number of results, sub-lists are used as queries to search the database. Sub-lists are used until enough results are returned to include all word-features. Each word-feature is covered no more than once, so that when a non-empty result is returned, all other queries containing any of those word-features are filtered out.

The program for generating sub-lists, while keeping the order of words in the original sentence, is as follows. Let S be all the non-empty sub-lists of the word feature list A, sorted by length, where S_1 is the complete word feature list, and $S_{|S|-|A|}, ..., S_{|S|}$ are the singleton features. Table 1 demonstrates an example of a sub-lists generated from Algorithm 1.

input : list A of word-features
output : set S of all sub-lists of A

```
// Create a window to view i number
   of elements
for i ← |A| to 1 do
  // Slide window along from left
  for j ← 0 to (|A| − i) + 1 do
  |  S_{|S|+1} ← A_j, ..., A_{j+i};
  end
end
```

Algorithm 1: Sub-list generator algorithm

The number of sub-lists generated using Algorithm 1 is always a triangle number, which given n word-features, is simply calculated:

$$\frac{n(n+1)}{2} \tag{1}$$

Upon generating the set of word-feature sub-lists, a database is queried with those sub-lists starting with the largest. SLiCE attempts to build mutually exclusive sets of recommendations R for each of the sub-lists s in S. A successful search returns a non-empty set of recommendations from the database entries B.

As expressed in Algorithm 2, if any elements of a sub-list s from S are in a previous result r in R, the sub-list is ignored. Otherwise, the sub-list s is used to query the database, and if the search returns one or more results, then the results b from B are added to a new index of R.

This process is exemplified in Table 2. The sorted queue of sub-list are used to query the Freesound database. Successful search results contain a set of one or more recommendations that each contain a file name, and URI.

Table 1. All sub-lists generated by SLiCE from a word-feature list.

order	query
1	hot summer days river hum insects
2	hot summer days river hum
3	summer days river hum insects
4	hot summer days river
5	summer days river hum
6	days river hum insects
7	hot summer days
8	summer days river
9	days river hum
10	river hum insects
11	hot summer
12	summer days
13	days river
14	river hum
15	hum insects
16	hot
17	summer
18	days
19	river
20	hum
21	insects

input : set of sub-lists S, and a set of database entries B
output : set of sets of audio file recommendations R

foreach s *in* S **do**
  if $\forall r \in R, s \cap r = \emptyset$ **then**
    if $\exists b \in B, s \subseteq b$ **then**
      $R_{|R|+1} \leftarrow b$
    end
  end
end

Algorithm 2: For each sub-list of word-features test if word-features from the current sub-list have been in a previous result. If not, then query the database for recommendations that contain the word-features in the sub-list. If a successful result is returned, those results are appended to the sets of recommendations.

3.4 Associated Tweet Word-features

Audio Metaphor creates a second set of recommendations based on Tweets. This is achieved by using the initial word-features from the natural language query to search for Tweets. Searching Twitter with the Twitter API [21] returns the latest thirty (30) Tweets from a query.

When Tweets are returned from a search, Audio Metaphor extracts word-features from the Tweets, removing common words, and leaving only nouns, verbs, and adjectives, as described in Section 3.2. Additionally, words from the initial search query are discarded. The remaining words are sorted by frequency of occurrence. Words of equal frequency are handled first-in first-out. The same number of words as was in the word-feature list from the natural language query are used to create the new word-feature list, in effect, doubling the space for Audio Metaphor to explore. The new word-feature list is input to SLiCE which generates a second set of recommendations.

Table 2. Query queue indicating the empty searches $\emptyset$, successful searches in bold, and ignored queries grey.

order	query	results
1	hot summer days river hum insects	$\emptyset$
2	hot summer days river hum	$\emptyset$
3	summer days river hum insects	$\emptyset$
4	**hot summer days river**	7
5	summer days river hum	-
6	days river hum insects	-
7	hot summer days	-
8	summer days river	-
9	days river hum	-
10	river hum insects	-
11	hot summer	-
12	summer days	-
13	days river	-
14	river hum	-
15	**hum insects**	13
16	hot	-
17	summer	-
18	days	-
19	river	-
20	hum	-
21	insects	-

In our example, after word-feature extraction from the thirty most recent Tweets, the top six most frequent words are:

1. song
2. hate
3. love
4. extractor
5. time
6. outside

Table 3 demonstrates the sub-lists generated by SLiCE in our example.

3.5 Audio File Recommendations

After Audio Metaphor retrieves audio file recommendations for a natural language query, two sets of recommendations are returned to the soundscape composer. The first set are those recommendations that are returned directly using the word-features from the natural language query. The second set of recommendations are derived from associated words found by searching Twitter. In different compositional situations we recognize that composers may want different number of recommendations, and Audio Metaphor provides a mechanism to specify the maximum number of recommendations returned in each set.

Table 3. Query queue indicating empty results ∅, successful results in bold, and ignored searches in grey.

order	query	results
1	song hate love extractor time outside	∅
2	song hate love extractor time	∅
3	hate love extractor time outside	∅
4	song hate love extractor	∅
5	hate love extractor time	∅
6	love extractor time outside	∅
7	song hate love	∅
8	hate love extractor	∅
9	love extractor time	∅
10	extractor time outside	∅
11	song hate	∅
12	**hate love**	2
13	love extractor	-
14	extractor time	∅
15	**time outside**	13
16	**song**	2251
17	hate	-
18	love	-
19	**extractor**	11
20	time	-
21	outside	-

For each set, Audio Metaphor will return no more than the number of audio files requested by the composer. Each set of recommendations may have a different number of subsets, with the number of recommendations returned for each subset equal; specifically the number specified by the composer is divided by the number of subsets within a set. The number of audio files in each subset returned to the composer is specified by this number, and if the number in a subset falls short of this number then it is ignored. The two modified sets of recommendations are then returned to the composer.

In our example, given that a composer desires ten (10) audio file recommendations for a natural language query and there were two subsets in the set of recommendations made using the word-features directly from the natural language query that had two subsets of successful search results there would be 5 recommendations made for each of those. The associated Twitter words returned four subsets of recommendations, resulting in three recommendations made for each of those after rounding.

4. IN[A]MOMENT

In[a]moment is a interactive performance piece [22] created under the eight-month Interactive Arts and Technology Capstone course at Simon Fraser University (SFU), School for Interactive Arts and Technology (SIAT). The project took place from September 2011 to April 2012, with its debut showcase on April 2nd, 2012 in the performance and technology show PIT 2012 [23] at SFU Woodward's Studio D in Vancouver. Furthering the success of this production, *In[a]moment* was chosen to be presented in the 2012 BCNET Digital Media Challenge [24].

Figure 2. Figure depicting the different actors and their interactions for *In[a]moment*. A person in the audience sends Tweets to Audio Metaphor. Audio Metaphor then provides the musician with audio file recommendations for soundscape composition. The dancers respond to the soundscape audio. The audience might then Tweet their experience response back to Audio Metaphor.

Through a combination of contemporary improvised dance and technology, *In[a]moment* explores the moments which occur when the interactions of the audience, performer, and space intersect. The performance environment was a darkened theatre space with a two channel audio configuration, and scrim (a semi-opaque material) hung from the ceiling to the floor as a projection surface. Stations were positioned to either side of the stage for technical production. The audio performer was at one station and an automated visual composer was at the other. Another computer was used to run Audio Metaphor.

This implementation of Audio Metaphor was as a performance based application system, where performers and audience members could interact through the social media network Twitter. The audience was instructed to interact with the system by posting Tweets that included @inamoment. Audio Metaphor listened for Tweets containing @inamoment during the performance and accepted these as input phrases for processing. Tweets were handled first-in-first-out, which update the sound file recommendations as Tweets were processed. Audio file recommendations were sent by wireless network to the soundscape composer, who combined and processed several of the audio file results for the soundscape composition. The composer in this case would also add other musical content for the performance.

4.1 Preliminary Results

The audience was invited to enter phrases through their personal mobile devices. These natural language phrases were then used with our algorithm to supply audio file

Figure 3. Image of a dancer with word-features projected into the environment.

recommendations to a human composer, who in turn wove the sounds into the soundscape performance.

There was between 150-200 people in the audience for each of the three performances, who together sent between 10 and 25 Tweets for each performance. The performance time was 12 minutes. The number of audio file recommendations for a natural language query was set by the composer at the beginning of the performance. Recommendations were updated as they were processed by Audio Metaphor and sent to the composer via the OSC protocol. To resolve delay issues associated with audio file download times, 2000 database entries, tagged as either field recording, or soundscape were mirrored from Freesound onto a local database. New recommendations were being made with a high frequency, and the system would notify the composer each time a new set of recommendations was available. The composer would decide when to update the set that he was working from to be the newest set. This push / pull set up gave the composer creative control allowing him to continue working from a set of recommendations until he felt it was an appropriate time to get a new set.

Evaluation of the system was through its application within the performance, and the success of the performance. The success of *In[a]moment* was measured by its acceptance into the BCNet Digital Media Challenge, and through casual interviews with audience members. Audience feedback indicated that the soundscape composition had representations persistent with those contained in the natural language phrase. The composer who used Audio Metaphor indicated that the audio files were appropriate for use in a compositional representation of the natural language query. The composer noted that the discrete selection of files, which were presented by Audio Metaphor, provided an effective working space for realtime soundscape composition.

5. CONCLUSIONS AND FUTURE WORK

We have described a system to address the problem of automatically exploiting online data from a high-level semantic perspective for soundscape composition. Natural language phrases are used to generate database queries to find a combination of search results for representing the phrase. Additionally, the search space of the phrase is enriched by generating new queries from related social media content. We have presented how social media can be used to generate related queries to enrich search results for realtime soundscape composition.

Audio Metaphor has been employed in the contemporary dance and technology production *In[a]moment* as a performance based application system. The application of the Audio Metaphor, as a performance application for *In[a]moment*, demonstrates a positive direction in using natural language and social media for audio information retrieval in real-time contexts.

In future work, we plan to develop Audio Metaphor with the ability to autonomously generate soundscape compositions. By using audio analysis features and durations we will endow the system with the capacity to create full compositional experiences for the lister. We would further like to utilize semantic technologies for better machine understanding of intention in user descriptions and tags. Furthermore we are working toward the classification of audio files for soundscape composition to aid in the composition decisions made by machines.

Acknowledgments

Acknowledgements go to members of the production *In[a]moment* Winnie Chung, Jon Bantados, Carol Tu, Jackie Ho, and Chao Feng. We would also like to acknowledge the production assistance of professors Thecla Schiphorst, and Henry Daniels from Simon Fraser University. This research was made possible, in part, by a grant from the *Natural Sciences and Engineering Research Council of Canada*.

6. REFERENCES

[1] S. Wilson, D. Cottle, and N. Collins, *The SuperCollider Book*. Mit Press, 2011.

[2] J. Kreidler, *Loadbang: Programming Electronic Music in Pd*. Wolke, 2009.

[3] V. Manzo, *Max/Msp/jitter for Music: A Practical Guide to Developing Interactive Music Systems for Education and More*. Oxford University Press, 2011.

[4] V. Akkermans, F. Font, J. Funollet, B. de Jong, G. Roma, S. Togias, and X. Serra, "Freesound 2: An Improved Platform for Sharing Audio Clips," in *International Society for Music Information Retrieval Conference*, 2011.

[5] T. O'Reilly and S. Milstein, *The Twitter Book*. O'Reilly, 2009.

[6] A. Eigenfeldt and P. Pasquier, "Negotiated Content: Generative Soundscape Composition by Autonomous Musical Agents in Coming Together: Freesound," in *Proceedings of the Second International Conference on Computational Creativity*. México City, México: ICCC, 2011, pp. 27–32.

[7] N. Finney and J. Janer, "Soundscape Generation for Virtual Environments using Community-Provided Audio Databases," in *W3C Workshop: Augmented Reality on the Web*, Barcelona, Spain, June 2010.

[8] H. Becker, M. Naaman, and L. Gravano, "Learning Similarity Metrics for Event Identification in Social Media," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, ser. WSDM '10. New York, NY, USA: ACM, 2010, pp. 291–300.

[9] E. Agichtein, C. Castillo, D. Donato, A. Gionis, G. Mishne, E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, "Finding High-quality Content in Social Media with an Application to Community-based Question Answering," in *In Proceedings of WSDM*, 2008, pp. 183–194.

[10] C. Castillo, M. Mendoza, and B. Poblete, "Information Credibility on Twitter," in *Proceedings of the 20th International Conference on World Wide Web*, ser. WWW '11. New York, NY, USA: ACM, 2011, pp. 675–684.

[11] A. Pal and S. Counts, "Identifying topical authorities in microblogs," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, ser. WSDM '11. New York, NY, USA: ACM, 2011, pp. 45–54.

[12] Stanza. Sound Cities. Available online at http://www.soundcities.com; visited on April 12th 2012.

[13] D. Holzer. Sound Transit. Available online at http://turbulence.org/soundtransit/index.html; visited on April 12th 2012.

[14] The British Library. Archival Sound Recordings. Available online at http://sounds.bl.uk/ProjectInfo.aspx; visited on April 12th 2012.

[15] L. Nie, M. Wang, Z. Zha, G. Li, and T.-S. Chua, "Multimedia answering: enriching text qa with media information," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information*, ser. SIGIR '11. New York, NY, USA: ACM, 2011, pp. 695–704.

[16] J. Wang, X. Chen, Y. Hu, and T. Feng, "Predicting High-level Music Semantics Using Social Tags via Ontology-based Reasoning," in *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010, Utrecht, Netherlands, August 9-13, 2010*, J. S. Downie and R. C. Veltkamp, Eds. International Society for Music Information Retrieval, 2010, pp. 405–410.

[17] J. Stockholm and P. Pasquier, "Reinforcement Learning of Listener Response for Mood Classification of Audio," in *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, ser. CSE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 849–853.

[18] Freesound 2 API. Available online at http://www.freesound.org/docs/api/overview.html; visited on April 12th 2012.

[19] E. Loper and S. Bird, *NLTK: The Natural Language Toolkit*. O'Reilly Media, Inc., May 2009.

[20] Oxford University Press, "The OEC: Facts about the language," Available online at http://oxforddictionaries.com/words/the-oec-facts-about-the-language; visited on April 12th 2012.

[21] Twitter API. Available online at https://dev.twitter.com/docs; visited on April 10th 2012.

[22] In[a]moment project website. Available online at http://iamoments.wordpress.com/; visited on April 12th 2012.

[23] Performance in Technology event website. Available online at http://www.siat.sfu.ca/pit2012; visited on April 12th 2012.

[24] BCNet. BCNet Digital Media Challenge. Available online at https://wiki.bc.net/atl-conf/display/BCNETCONF2012/Cast+Your+Vote+for+Student+Innovators+in+Digital+Media; visited on May 2nd 2012.