An Evaluation of Comparison Generation in the Methodius Natural Language Generation System

Matthew R. Marge

Master of Science

Artificial Intelligence

School of Informatics

University of Edinburgh

2007

Abstract

This thesis performed an evaluation of the Methodius Natural Language Generation System's parameterized comparison generation algorithm. For our study, we generated texts about music using the Methodius Natural Language Generation System. In order to gain a sense of what disc jockeys discussed about music pieces, we transcribed a number of disc jockeys from a classical and jazz radio station. We then authored a knowledge base of facts about music pieces based on the types of facts disc jockeys frequently discussed. We conducted an experiment to test several hypotheses that evaluated comparison generation in Methodius. To accomplish this, we developed and executed a web experiment where participants read a number of paragraphs and answered actual recall questions about jazz and classical music pieces. The primary purpose of our experiment was to test whether people learned more from texts containing comparisons produced by Methodius versus texts that did not contain comparisons. Our results confirmed this hypothesis. These results also verified that Methodius' parameterized comparison generation algorithm could generalize to the music domain.

Acknowledgements

I would first like to thank my supervisor Johanna for her support and guidance throughout the entire thesis process: defining a topic, deciding upon the necessary steps to produce the thesis, and allocating the resources to complete it. Much thanks to Amy and Colin for their comments on my thesis, for familiarizing me with state-of-the-art natural language generation systems, and guiding me throughout all the bugs and detours that we encountered to produce results from Methodius. I would also like to thank Dr. Frank Keller and Neil Mayo for their crucial advice while developing and hosting a web experiment using WebExp. Thanks to my past instructor Dr. Ellen Bard for helping me through the web of potential data analyses. Thanks are also in order for Keith Edwards and Ray Carrick for their helpful comments and suggestions for my experimental design.

I would like to thank my family for their support and for the encouragement to study abroad in the homeland of my grandfather James' parents James and Catherine. I am in deep gratitude for all the lifelong friendships I made while at Edinburgh. We all helped each other navigate through the intense course of study that is the MSc in Informatics at Edinbugh.

Thanks also to the participants of my study and to the All Music Guide for providing the music data necessary for the development of my experiment.

Lastly, I would like to thank the Saint Andrew's Society of the State of New York and the Edinburgh-Stanford Link for funding my studies. It was a life-changing experience (for the better)!

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Matthew R. Marge)

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Importance of Comparisons in Natural Language Generation

There have been several recent works in support of tailoring natural language to a user's previous browsing history in domains such as medicine, museum collections, and animal descriptions [1-4]. An upcoming project funded by the Edinburgh-Stanford Link, "DJ4me", is a proposed digital music player that will feature a user's own personal disc jockey (DJ) [5]. The purpose of the DJ is to discuss interesting trivia or facts about songs recently played to the user. To date, there has not been a form of user modeling for this application, which could benefit from customization. Commercial music applications such as Last.fm provide users generic information about an artist as the artist's song is being played on the user's music player [6, 7].

Natural language comparisons (including comparisons and contrasts) between music artists or songs could provide users with a novel way to explore their music collection. Furthermore, when compared to plain text descriptions, previous research has shown that people tend to discover more and feel that they discover more information from text that is enriched with comparisons and methods to combine facts into fewer sentences (i.e., aggregations) [8]. The Methodius Natural Language Generation (NLG) system makes this text enrichment possible by forming customizable descriptions of objects from a defined database. Methodius features a novel algorithm for generating comparisons between objects that are currently being encountered and those that have previously been encountered. This comparison-generating algorithm stands out from previous attempts because it chooses the most relevant and interesting comparisons given a context that is set by several explicit parameters. These parameters set the degree of importance of the following factors:

A) Comparisons are preferred to be between larger groups as opposed to smaller ones.

B) Comparisons are preferred to have as many facts to compare as possible.

C) Comparisons are preferred to be between objects that are similar.

D) Comparisons are preferred to be between recently encountered objects.

Currently, the Methodius project generates natural language descriptions of cultural artifacts [9].

1.2 Problem

The comparison algorithm featured in Methodius has not yet been proven to generalize across domains. Therefore, we propose to collect and prepare the necessary data to assess whether this parameterized comparison algorithm can be successfully applied to the music domain. The text to be evaluated will feature facts about music artists and their songs. The parameters of the comparison algorithm will be set based on collected transcriptions of human disc jockeys discussing music facts between songs. We will evaluate whether text generated using this novel comparison approach is more informative and is perceived to be more informative to human users than text without comparisons by performing a user study.

1.3 Hypotheses

Main hypothesis: People will retain more information from text generated using Methodius' parameterized comparison algorithm then text generated without comparisons.

Supplemental hypothesis: People will perceive that they learn more from text generated using Methodius' parameterized comparison algorithm than text generated using Methodius without comparisons.

Additional hypotheses: People will find text generated using Methodius' parameterized comparison algorithm to be more interesting and enjoyable than text generated using Methodius without comparisons. The comparison algorithm generalizes to the music domain.

Following this investigation, we can conclude whether Methodius' parameterized comparison algorithm will be appropriate to feature in the "DJ4me" project. If the text generated using the parameterized comparison algorithm is shown to be more informative than text generated without comparisons, we believe it will enhance users' experience with music played using "DJ4me". In addition, if this study shows that the comparison algorithm generalizes to a new domain, the comparison algorithm could be used in many similar applications, including online shopping websites, airline reservation systems, and restaurant recommendation systems. This investigation will also help us better understand the importance of comparisons in generated texts in the music domain.

Chapter 2

Background

2.1 Previous Approaches to Generating Comparisons

2.1.1 Introduction

Natural language comparisons between objects provide users with an additional way to explore a knowledge base. Comparisons can be made to relate an object to similar objects that have previously been described. These factors increase a system's perceived intelligibility by users [10].

Approaches to generating comparisons began two decades ago with the TEXT system [11]. Until recently, most systems largely generated comparisons in limited ways, such as with templates. Comparison generation algorithms have gradually increased in sophistication as data structures composed of potential comparators became more organized. The most recent approach includes customizable parameters that when altered, tend to make the NLG system produce different types of comparisons between objects [9]. We hypothesize that systems that permit customizations of comparisons via parameters will be most effective at generating comparisons based on a user's interaction history with an NLG system.

Our focus in this review section is on the TEXT system and systems that do not require the user to request a direct comparison between objects. We will describe their approaches to generating text that compares objects from a knowledge base Next, we will assess how these systems form comparisons, if there are specific requirements that must be met. NLG systems will also be evaluated for their use of discourse history and other related information in order to observe if their resulting texts are contextually appropriate. As a general comparison among these systems, we will describe an overview of each system's pipeline for generating text. We will compare each system to the traditional generation pipeline proposed by Reiter and Dale [12]. Nearly every system we will discuss has only been applied to one domain. As an evaluation for each system, we will explore each system's ability to be applied across domains.

2.1.2 The TEXT System

The TEXT generation system, developed in the early 1980's, is one of the first text generation systems to provide comparisons between items from a knowledge base [11]. In this system, the knowledge base consists of military hardware. This system generates text based on *schemata*, software tools that provide guidelines for presenting a specific selection of facts in natural language. There are four types of schemata in the TEXT system. Based on an input question, TEXT selects one of these schemata and collects a "relevant knowledge pool" of information that could be included in its response [13].

Comparison Generation

The *compare and contrast* schema is responsible for generating comparisons between two objects in a knowledge base. A comparison between the objects is generated based on the common attributes of both objects. This comparison employs only the *identification* schema to this group of objects, which generates information about the group's common attributes and hierarchical classifications (if any exist) [11].

After selecting the *compare and contrast* schema, this schema selects one of the other schemata (*identification, attributive, constituency*) to contrast both objects. TEXT selects the most ideal "supplemental" schemata by measuring the degree of similarity between both objects. The degree of similarity between the two objects to be discussed is then calculated based on the meaningful information TEXT has on both objects [11]. When the objects to be discussed are very similar, TEXT selects the *attributive* schema because detail about both objects is available for access. However, when the objects are very different, TEXT selects the *identification* schema because little information about both objects is accessible to the system. Instead, information about their relationship to the knowledge base's object hierarchy is provided. If a comparison is to be generated between two objects that are neither very similar nor different, then TEXT selects the *constituency* schema, which describes a mix of attributive information (i.e., features of the two objects) and taxonomic information about both objects [11]. In order to generate text that contrasts the two objects, the *compare and contrast* schema is executed twice (once for each object). The last sentence of the resulting text from executing the *compare and contrast* schema is a straightforward comparison between the two objects.

A critique of this approach is that schemas bind the system into grouping database facts into either (1) attributes of objects, or (2) their classification in the military hardware hierarchy. Other information about the database objects, such as distinctive information about the time period of an object, cannot be clearly distinguished, especially when the system attempts to differentiate between

two objects in the same location in the hierarchy and in the same relevant knowledge pool [11]. Customization of generated responses is also not permitted. Comparisons are only generated when explicitly asked for by the user. Thus, comparisons between the current object being discussed and a group of objects are not permitted.

System Architecture

The TEXT system served as a foundation for traditional NLG pipelines. When a question is input to the system, TEXT pools together information from the knowledge base that it considers to be appropriate for a response. It then selects a schema type for its response based on this "relevant knowledge pool." Next, it "fills in" the schema by using the schema's semantic properties to select the most appropriate knowledge to generate in a response. If there are multiple appropriate knowledge options, a "focus" component selects knowledge that best follows the most recent discourse. The system then uses this "filled schema" to generate natural language text [11, 14].

Context Representation

To maintain context, TEXT uses the most recent discourse. However, the history is only one level deep. After a question is entered, TEXT only refers to the preceding question for maintaining "focus" [13]. Thus, the system can repeat information already presented to the user. In [11], the author discusses the potential for tracking a sophisticated interaction history that is addressed in future research.

Potential for Applicability across Domains

The TEXT system could be applied to other domains (e.g., music, animal classification) if they follow a similar hierarchical makeup to military hardware. To use TEXT for other domains, one only needs to adapt the access functions that bind abstract rhetorical predicates to the new knowledge base.

2.1.3 The Migraine System

Another NLG system that served as the foundation to those found today is the "Migraine" system. This system, intended to help users of migraine medication, generates text from a knowledge base of medications. It operates much like TEXT in that it generates responses to queries based on user input.

System Architecture

This system also precedes the establishment of a traditional generation pipeline. If a user enters a query about medical information, the Migraine system first establishes a *communicative goal*. This goal must be achieved by the system in its response to the user. The *text planner* then uses this goal to retrieve operators that are to be used to build a plan for generating an appropriate response. These *plan operators* are selected based on several conditions, including the user's knowledge and the appropriateness of the operators given their requirements. Using the selected operators, the text planner builds a tree that maps out the explanation that will be given to the user. A response is then generated with knowledge that best answers the user's query [1].

Comparisons and Context

In addition to responding to the user's query, the Migraine system maintains a *discourse history* of the overall interaction with the user, including previously mentioned medications. This system generates comparisons between migraine medications by accessing previously provided medical information via its discourse history. When a new medication is to be described, the system searches its knowledge base using selected plan operators for ways to compare and contrast the requested medication to those previously mentioned. For example, one plan operator type, when selected, mentions the similarities between the current medication and those previously described. Another plan operator type mentions the benefits of the current medication over those previously described. There are very few restrictions placed on the types of comparisons that can be made between medications. Future research will improve restrictions on comparison generation in order to improve the perceived importance of comparisons, and prioritize those which are seen as most important for the user [10].

The system follows Rhetorical Structure Theory (RST) by expanding plan operators into nuclei and satellites [15]. For each plan operator expansion, the *nucleus* region of the operator is first executed, which may include intentions to compare the current medication to previously mentioned ones. In addition, the plan operator may then execute its *satellites*, which expand the plan operator further by including *subgoals* that permit the system to elaborate further about the similarities and differences between medications [1]. The system compares and contrasts medications and their respective side effects. This allows the user to interpret how medications are related. The system restricts the level of detail (e.g., explaining medical terminology) during comparisons based on its assumption of what the user already knows. The system accesses this information from its discourse history and the model of the user.

The system's ability to compare and contrast medications is dependent on plan operators. This means that the variety of comparisons is limited to those explicitly written. However, the

system's ability to generate comparisons based on this limitation does allow them to tailor to the medical domain, which requires special information such as side effects.

When the user asks a supplemental question about a previously mentioned medicine, the Migraine system accesses its log of the previous discourse to generate the most appropriate response. When the user asks a question such as "Why?", the system accesses the past and present discussed medications to explain its response. However, if the system assesses that the user is already aware of this explanation, it attempts to locate a preceding proposition that was provided. When the user asks for clarification (i.e., the question "Huh?"), the system searches its discourse history for any gaps in achieving its communicative goals [1]. Also, the system ensures that no communicative goal has already been achieved previously by accessing its previous discourse.

Potential for Applicability across Domains

Due to the limitation of technology at the time, the authors developed plan operators specifically for the original application to the medical domain. Unfortunately, this does not permit the system to work well in other domains, where information such as side effects and patient restrictions are not present. Significant modifications to system software would be required to apply this system to other domains.

Part II: Hypertext Generation Systems

2.1.4 PEBA-II

The PEBA-II system provides a method of generating comparisons that has been widely adopted in today's NLG systems. It is the first hypertext generation system we will discuss. Intended for educational purposes, this system generates descriptions of animals and comparisons between them from a knowledge base. Unlike previous systems, the comparisons generated can either be directly requested or implicitly provided, i.e., the user does not need to directly request them. Instead, the system generates comparisons between animals as the user navigates a *hypertext* interface similar to that of an electronic encyclopedia. For each page that the user views, the system produces comparisons between the current observed animal and other animals. The types of comparisons that the system makes are tailored to the user's expertise as either a novice or an expert [4].

Comparison Generation

There are three types of comparisons featured in the PEBA-II system. Whenever PEBA-II generates a comparison, the current animal being described will always be included, unless a *direct comparison* is made. A *direct comparison* presents the similarities and differences between two user-

specified animals. The user typically requests direct comparisons, except when the system finds that multiple subcategories of a class of animals need to be clarified (e.g., generating a distinction between breeds of goats). Similarities and differences between two animals are presented in order of perceived relevance to the user [4].

Illustrative comparisons clarify a feature of the animal being described by relating the feature to a similar one of a previously mentioned animal. For example, the statement "goats have deerlike horns" is a type of illustrative comparison. This type of comparison is generated by the system when the system deems the fact to be worth mentioning [4].

Occasionally, animals that were already mentioned can be easily confused with the current animal being described because of their similarities. A *clarificatory comparison* alleviates this problem by making a distinction between the current animal being described and one with which it could easily be confused (e.g., similarities and differences between jaguars and cougars). The system also generates a clarificatory comparison when it finds key similarities between the current animal being described and one that was previously described. Then, PEBA-II selects similarities and differences for the clarificatory comparison that it determines will best educate the user [4].

Clarificatory comparisons stand out from the other two types of comparisons because they implicitly mention the key similarities and differences between two commonly confused animals. Similar to the TEXT system, PEBA-II occasionally contrasts animals based on their classification in the animal hierarchy [11]. However, this is not always the case. Often PEBA-II elaborates on the visually observable features to make a distinction between two animals. This is particularly true when the system classifies the user as a novice. The author believes it is crucial that comparisons are not dependent on an animal's hierarchical classification. To determine the importance of mentioning a similarity or difference, the author pre-ranks the attributes of animals with weights. These weights vary depending on the expertise of the user. For novice users, visually observable features are ranked higher than other biological properties. Expert users are believed to prefer the "internal" features of animals [4]. Thus, the system will adapt to its user. However, it is limited to only these two classifications; the user sets no specific parameters. A *true* expert user would have to be an expert in all types of animals. Most users would need to thereby be classified as novices for the system. This generality tends to produce issues with users who are experts in some aspects of animal biology. We expect that the system will be perceived as more effective if it tailors generated comparisons based on several user-specified preference parameters. This will be addressed in subsequent research.

The system does not solely depend on the ranking of animal facts for clarificatory comparisons. For coherence, the selected attributes for the animal being described must match the attribute types of the other animal. Milosavljevic defined several frequent attributes from her corpus analysis of animal texts. For similarities in clarificatory comparisons, she found that *appearance, size,* and *taxonomic relationship* (i.e., similarities between the hierarchical classification of two animals) to be most common, along with rare features of the animals. These rare features are predetermined by the author, and most likely follow the opinion of the author. Different people find different features of animals to be exceptional. When highlighting the differences between animals, she found that *size,*

taxonomic relationship (i.e., the differences between the hierarchical classification of two animals), *shape,* and *body covering* to be most common. Distinctive facts about one animal that are not present in the other are also commonly mentioned as key differences [4].

PEBA-II's process of generating a clarificatory comparison requires that two animals must be very similar. For instance, they may be on the same branch of the animal hierarchy. The system determines the degree of similarity between two animals by whether they share key properties that are not found in other animals. This is not clearly defined, and could very much depend on the opinion of the person authoring the knowledge base. The author classifies this as an avenue for future research. When the system attempts to generate a clarificatory comparison between two animals, it first searches the pre-ranked knowledge base for important similarities between the two animals. Next, the system selects features that differentiate the two animals that are strongly related to their key similarities, if any exist. Then, PEBA-II selects the most important features labeled as "typical" for differences between the two animals (i.e., *size, taxonomic relationship, shape,* and *body*). When the animal *not* being described on the hypertext page possesses a feature that distinguishes it from the current animal being described, the system includes it in the comparison (e.g., "Goat, common name for any of eight species of cloven-hoofed, horned mammals closely related to sheep. The two differ in that the goat's tail is shorter and the hollow horns are long and directed upward, while those of the sheep are spirally twisted.") [4].

System Architecture

The PEBA-II system follows the pipeline architecture of most traditional NLG systems [4, 12]. It is extended to function as a resource that is accessible via the Web. A *user model* maintains whether the user is a novice or expert of animals. The system also keeps track of all animals that are well known by the user in the user model. The system possesses two types of *discourse plans*, which operate like McKeown's *schemata* [4, 11]. The *identity* discourse plan provides a plan for describing an animal. This plan generates *clarificatory* and *illustrative comparisons*. When a *compare-and-contrast* discourse plan is posted, the system generates a *direct comparison*. Since PEBA-II is a hypertext environment, the authors emphasize that a user's traversal of hyperlinks is a type of discourse, which is stored. Discourse plans are realized into hypertext and presented in a web browser [4]. If any pictures of animals are included in the animal description, they are included in the web browser.

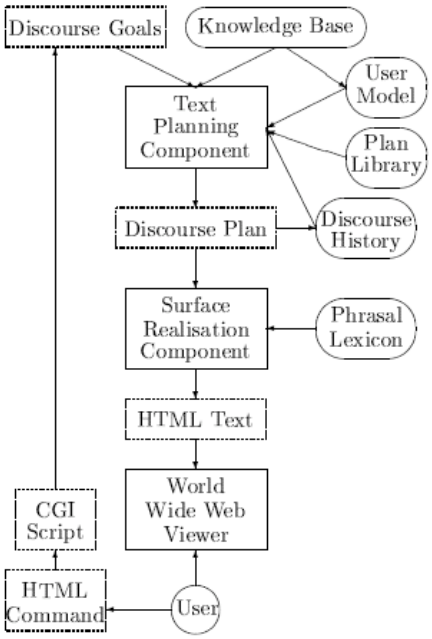

Figure 2.1: The PEBA-II system architecture. [4]

Context Representation

The system takes advantage of the hypertext environment by tracking the pages the user selects during an interaction. This *discourse history* is used to highlight key similarities and differences between the current animal being described and other animals [4]. It also ensures that facts are not repeated during an interaction.

Potential for Applicability across Domains

Since the knowledge base is handwritten specifically for the animal classification domain, it cannot be applied across domains. The PEBA-II software, however, as shown in the initial version of the POWER NLG system, contains rules that are designed independent of its original intended domain [2]. Modifications (including the construction of a new knowledge base) would be necessary for this system to migrate to a new domain.

2.1.5 POWER

The POWER system serves as a follow-up to PEBA-II. It is intended to be a multilingual information resource for museum objects. Unlike PEBA-II, the knowledge base is constructed automatically from an existing database. As a first cut, they modeled their NLG system off of PEBA-II [2, 4]. They were able to successfully construct a draft of POWER through minor modifications to PEBA-II and the implementation of a small hand-authored knowledge base.

Comparison Generation

Comparisons produced by the POWER system are generated in the same manner as PEBA-II. On each museum description webpage presented to the user, the authors included a feature to directly compare the current object being described to one of a list of other museum objects. The lexicon for generation was largely obtained from the existing database. Non-English languages required a manual translation of words. The limitations for comparison generation that hold for PEBA-II also hold for POWER. They concluded that generated texts, including comparisons, from the automatically constructed knowledge base were much more basic than those produced from the handwritten knowledge base [2]. This result is to be expected unless, as they argue, databases are more carefully structured to tailor to NLG applications. Future research addresses this issue partially by producing an authoring tool that lets novice users compose a rich knowledge base for a generation system [16].

System Architecture

The pipeline for POWER is almost entirely based on the previous PEBA-II system, except for the creation of a knowledge base for the museum domain. Unlike PEBA-II, however, its surface realizer contains a grammar that specifies the syntactic structure of natural language text. The POWER system uses handwritten templates to denote knowledge about the discourse and grammar. The authors argue that templates' generic properties allow them to be applied in a variety of NLG research applications. Future research should address this issue because templates limit the flexibility of an NLG system's output to those a person explicitly composes. When a user selects one of the hyperlinks displayed on a museum description webpage, a new discourse goal is formed, which restarts the pipeline [2].

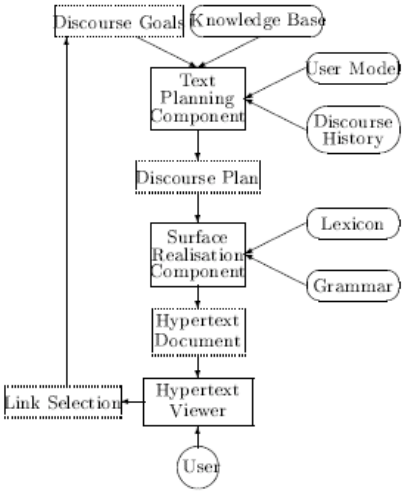

Figure 2.2: The POWER system architecture. [2]

Context Representation

POWER's representation of context is equivalent to that of PEBA-II. Thus, it also incorporates a history of user-system interactions [4].

Potential for Applicability across Domains

The database-driven version of the POWER system can certainly be applied to multiple domains. However, this will require the modification of the information extraction tools used to produce the knowledge base. The hand-authored version of the POWER system carries the same limitations as PEBA-II.

2.1.6 ILEX

ILEX (the Intelligent Labeling Explorer) produces natural language descriptions of jewelry and was a predecessor to Methodius. One application that uses ILEX is a web interface that allows users to browse pieces of jewelry one at a time. However, text on each page is not static. Text generated by ILEX will *dynamically adapt* to the user's browsing history. This means that whenever the system is executed, a web of linked documents is created that will be accessed by the user during the interaction. Each webpage contains hyperlinks to pieces of jewelry related to the object being described [3]. Whenever the user selects a hyperlink, all text about a piece of jewelry is automatically generated and presented in the user's web browser.

Similar to the Migraine, PEBA-II, and POWER systems discussed thus far, ILEX implements a user model to tailor generated text given the user's discourse with the system thus far. However,

unlike these systems, ILEX has its own specified *agenda* of when to present information to the user without knowing when the interaction will end. ILEX must accomplish this agenda as it interacts with the user. As an improvement over PEBA-II and POWER, ILEX possesses ability to generate text automatically without templates [3].

Comparison Generation

Unlike previous systems, the user cannot request a direct comparison between two objects. This type of feature should be included in this domain, since a user may need to compare the currently viewed product to a familiar one. No comparisons can be made for the first object presented to the user [3].

For each webpage, ILEX selects and presents a set of hyperlinks to objects that are related to the current object. When the user selects a related piece of jewelry, the generated text on the new webpage includes an indication of whether or not the piece is of the same style as the previously viewed piece. This comparison is implemented by inserting the word "also" into a sophisticated rhetorical structure that generates the description. If a third consecutively related item is selected, the phrase "the previous item" is added to the description [3]. Like museum object descriptions, ILEX includes information about general properties of jewelry. These can be used in comparing the object being presented to ones of similar type.

When the system determines that an insufficient number of descriptive facts can be presented to the user, it generates additional comparisons. In effect, comparisons are considered to be "backup" generation information that will not always be included in a jewelry description. Comparisons should be made for each generated webpage beyond the first because they allow the user to relate learned information to a currently presented object. These comparisons are between the current object being described and one that it determines to be *similar* to the current object. They use Milosavljevic's approach to measuring similarity between two objects [17]. ILEX also keeps track of fact types, which are helpful for determining the appropriateness of facts for comparisons. Once the system selects the second object for a comparison, it selects facts that compare and contrast the object. These facts are selected in part because of their type because fact types determine the similarities and differences between the two objects [3]. Like previous systems, comparisons between the current object and related groups of previous objects are not possible.

Knowledge Base Structure

The knowledge base is hand-constructed from objects in a relational database. For ILEX's primary domain, it obtained information about jewelry from a larger database of museum objects. These entries also held information that established relations between pieces of jewelry, including how jewelry should be classified hierarchically. Another type of stored relation was a *predicate*

definition for each entry. This stored information about the way in which the entry should be described and compared with other pieces of jewelry. It also included data for the user model, such as the perceived importance of the entry by the user. Generic facts about each piece of jewelry, such as its style, are also stored in the predicate definition [3]. Unfortunately, the lack of an authoring tool meant that only coding experts could write up the knowledge base. Most museum curators would be unable to build the knowledge base, thereby decreasing the potential for the richest comparisons possible.

During the text generation process, ILEX structures the knowledge base in the form of a *directed, acyclic graph*. There are three types of nodes used for this graph. *Entity nodes* represent either pieces of jewelry or classes of jewelry. Lines connect these nodes to represent the relations between them. *Fact nodes* hold *predicate definition* information for each entity, including the perceived importance of the represented fact. *Relation nodes* hold links between facts, including whether linked facts compare to each other or contrast each other. These nodes also follow Rhetorical Structure Theory by containing either two *nuclei*, or one *nucleus* and one *satellite* [15]. A *content potential* is generated once all relation, fact, and entity nodes are merged into one interconnected graph. Although most of the content potential is pre-created before runtime, comparisons are generated live. This decreases the speed of the system, but places less restrictions on the types of comparisons that can be generated [3].

System Architecture

The ILEX pipeline partially follows the standard pipeline for NLG systems, and is largely adopted by modern NLG systems [9, 10]. Before the system interaction, the *content potential constructor* generates a directed, acyclic graph of information about jewelry from the knowledge base.

Once the ILEX system is online, the *text planner* produces a high-level plan of generated text about the selected piece of jewelry. It also arranges the plan into a rhetorical structure modeled off of the *relation nodes* found in the content potential. ILEX only includes the relation nodes that correspond to facts mentioned in the text plan. The *text planner*, unlike previous systems, does not use *schemas* modeled after McKeown's TEXT system [11]. Instead, they approach text generation *opportunistically*: ILEX accesses the existing interaction history in order to devise a highly relevant text plan. A highly relevant text plan is one that details the globally focused piece of jewelry or provides background information on the jewelry description. However, these facts must have a high *degree of relevance*, i.e., facts must be closely related to the globally focused object in order to be relevant. Future research should investigate the efficiency of this "relevance-scoring" approach compared to competing approaches for text planning. When the facts relating to the globally focused object are in short supply, the system generates comparisons between the current object and similar objects [3]. Comparisons should be included initially because they provide another method of relating

the focused object to similar ones in the knowledge base. Most ILEX-generated comparisons should provide new, appealing information to the user.

Unlike traditional NLG systems, the sentence realizer and noun phrase realizer call each other during the generation process. Thus, the system is less modular than originally intended. They fill in the text plans with appropriate *clauses* and *noun phrases*. Clauses refer directly to facts that will be included in the webpage description of the selected piece of jewelry. Noun phrases refer to objects in the content potential that have "complex" structure [3].

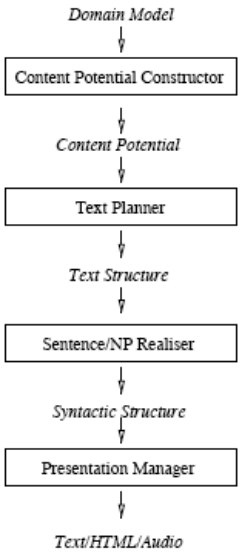

Figure 2.3: The System architecture for ILEX. [3]

Context Representation

Context for ILEX is represented in three ways: (1) a *discourse history*, (2) *local and global focus*, and (3) a *user model*. The ILEX system treats all user-selected hyperlinks as a type of *discourse history*. Like the Migraine, PEBA-II, and POWER systems, ILEX keeps track of the facts that have already been mentioned to avoid repeating information. ILEX also exploits the discourse history to vary sentence structure and discuss previously mentioned objects. The discourse history is also necessary for ILEX to make comparisons between the currently viewed object and those previously mentioned. Given a set of sentences to be generated, *local focus* represents the focused noun of the preceding sentence. *Global focus* holds the focused noun of the current jewelry page. These foci help place pronouns into generated sentences.

The *user model* holds weighted parameters about the perceived awareness (i.e., how likely the user is to be aware of a given fact) and importance of facts [3]. However, the representation of the user model is flawed in that it does not follow a sound metric that indicates the degree of perceived awareness or interest of a given fact. The authors manually entered initial scores for these parameters.

These scores are dependent upon the opinions of the authors. However, we support their modification of the awareness score based on the number of times the system presents the corresponding fact. To solve this problem, we suggest that the authors conduct a user survey to determine the types of facts that a critical mass finds to be well known or interesting. In addition, the user model is erased at the end of the interaction. If a user returns to interact with ILEX, the user model will have to be set again. M-PIRO, an enhanced NLG system largely based on ILEX, will address this issue by storing user modeling information on a remote server [16].

Potential for Applicability across Domains

ILEX was demonstrated to apply across domains. These domains included human resource management, computer-related products, and jewelry. However, one fallback to their approach is the need for a technical expert to enter information into the knowledge base. Future research will address this by including an intuitive authoring tool with an updated version of the system [10, 16]. The system also featured many language-dependent limitations in its components. This constrains its ability to apply across domains flexibly without requiring significant software revisions. The M-PIRO system will also attend to this issue [16].

2.1.7 M-PIRO and the Exprimo NLG Engine

The M-PIRO (Multilingual Personalized Information Objects) project serves to advance the state of the art in adaptive text generation by improving upon ILEX. Similar to ILEX, it is interfaced by a website where users can explore artifacts of ancient Greece in multiple languages. Each webpage holds one artifact and a brief description of the artifact. M-PIRO's generation engine, Exprimo, features the ability to vary the intricacy of artifact descriptions and control the number of facts presented in a generated sentence [16]. Both ILEX and Exprimo generated descriptions using the object's type, such as "the bracelet" as a reference to the current object [9].

Comparison Generation

Unlike previous NLG systems, Exprimo can compare the currently encountered object to either the previous object or to a group of objects that share its type. In addition, Exprimo can compare the currently encountered object to groups of objects recorded in the user's discourse history. Thus, relevant contrasts between objects can also be described to users. The authors explain that comparisons with objects beyond the preceding one should not be generated because they contain information that is likely to confuse the user. Instead, the system uses the discourse history to cluster together objects of similar taxonomical classification for a clustered comparison with the current object being described (e.g., "Unlike the previous classical period artifacts, this is from the archaic

period."). Comparisons may also be made with objects known to be resonant with the user by only mentioning the object's name [10]. Unfortunately, the knowledge base author is the person determining the resonance of objects. We believe that the user should be able to rate an object's resonance by explicitly describing the uniqueness of the object.

Exprimo generates two forms of comparison. A traditional comparison can be made between objects of the same type (e.g., *vessels*). Exprimo can contrast the currently viewed object with only a *cluster* of previously encountered objects. The cluster must belong to a classification on the museum object hierarchy that also corresponds to the currently viewed object. This information is accessed from the system's discourse history [10].

Exprimo must now select a set of possible comparators to the current object. The authors first created an offline set of appropriate ways in which to describe facts. One possible flaw is that this set is subjective and could vary from user to user. For instance, they find that numerical comparisons, such as differences in two objects' heights, are less important than the time period they were made. This may not be true for all users. When the system intends to form a comparison, it begins by building a taxonomical tree containing previously mentioned objects that share the same object type on the hierarchy as the current object, such as *vessels*. Each node on the tree contains either a museum object or a class to which it belongs. Also, each node includes two types of counts. An object count describes the number of previously mentioned objects that belong to each node, as shown in Figure 2.4 [10]. Several fact counts, also included for each node, describe the number of fact types that exist in that node's children. Each fact represents a possible comparator. In the Figure 2.4, for example, consider the current object to be a type of *lekythos*. The dashed arrows in the hierarchy point to the classes above and below the current object.

There are several discrete steps that Exprimo performs to select the most appropriate comparator, if any, from all facts contained in the tree shown in Figure 2.4. The system first removes each fact that has the following two properties: (1) the fact's index is '1', meaning that the fact describes only one object, and (2) the fact's type is not shared by the current object [10]. Since contrasts are only made between the current object and a cluster of objects, this step does not adversely affect the number of potential contrasts Exprimo can make. Instead, this step is effective at removing irrelevant contrasts.

Next, the authors attempt to improve the rate of general comparisons with the current object, which they argue is preferable to exceptionally lengthy and distracting detailed comparators. To accomplish this, Exprimo deletes all facts whose count is lower than the count of the class to which they belong. For example, in Figure 2.4, all facts in the *vessel* node are removed [10]. Unfortunately, this step limits the number of potential comparisons. Future research should also incorporate a user preference setting that will indicate if the user prefers to hear about specific fact types in comparisons. If any facts remain, Exprimo removes facts that are similar to those just deleted. Also, the system eliminates any facts that are not similar to the current object. However, the system keeps the remaining facts that are only mentioned in one previous museum object, if any exist. All remaining facts from nodes connected by dashed lines in Figure 2.4 are also kept because they are directly

related to the current object [10]. Although this approach is effective at restricting the number of potential remaining comparators, some removed facts might be desirable to the user. Future research will address this by breaking the selectiveness of this step into parameters [9].

At this stage, only a few possible comparators should remain for a comparison with the current object. The system now checks the remaining facts for those containing equivalent predicates. For example, consider our working example where the current object is a type of *lekythos*. Two facts of type *classical-period* may still exist, such as for the *lekythos* node (a *superclass* of the current object) and the *white-lekythos* node (a *subclass*). In our example, Exprimo eliminates the subclass fact because its fact count is lower than that of the superclass. If this were not the case, Exprimo would eliminate the superclass fact [10]. This process lends itself to an inherent preference to generic comparisons, which may not always be appropriate. Weighted parameters are one possible improvement to tailor this technique to the user's preferences. For example, a greater weight toward subclass facts would yield more specific comparisons with the current object.

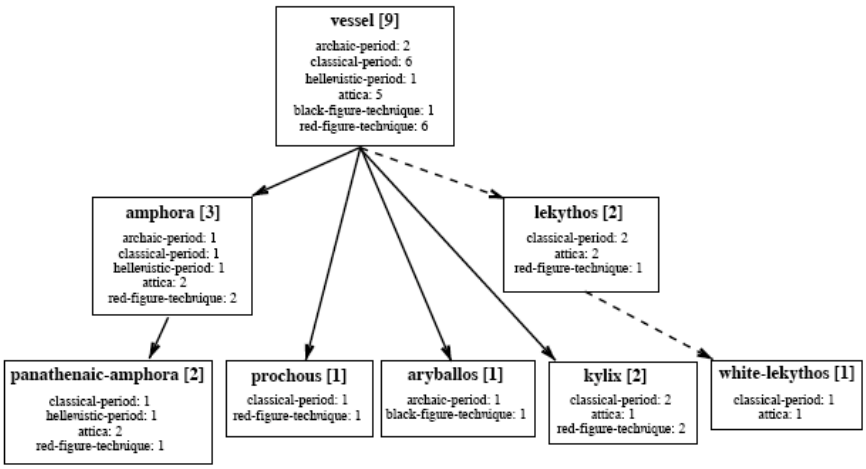

Figure 2.4: Tree structure used to form Exprimo comparisons. [10]

The final step to this process removes any remaining facts that do not share the same object type, subclass, or superclass with the current object. Exprimo now randomly chooses among any of the remaining facts to include in its comparison with the current object. The system produces a comparison emphasizing the similarity between the two objects if they share the same fact type (e.g., *both objects are from the classical period*). Otherwise, the system produces a sentence contrasting the differences between the two objects [10]. Unfortunately, it is possible that the system may need to randomly choose whether to compare the two objects or contrast them. To address this issue, the authors should survey system users in future research to observe each user's perceived level of interest of similarities versus differences when the system could generate both.

Knowledge Base Structure

Exprimo's *domain model* serves as the knowledge base and holds two components. Its structure was modeled off of ILEX's knowledge base. A *domain database* stores museum objects and the classes to which they correspond. This component also contains object *predicates*. Specifically, predicates hold object features and define relationships between objects. A *domain semantics* element provides specifications that restrict how the database can be examined by the system. Like ILEX, this component holds *predicate information* about each object, consisting of information about the way each entry should be described and compared with others. In addition, the hierarchical makeup of museum objects is stored in the predicate information [10].

Unlike previously mentioned NLG systems, Exprimo features an *authoring tool* that allows a more general audience to build M-PIRO's knowledge base. The authoring tool permits an author to enter museum objects and descriptions through an intuitive interface. Then, the tool automatically generates a knowledge base in XML format. In addition, the author can set *user model* parameters such as the perceived importance of an object [10]. This tool addresses a key issue in advancing the state of the art of comparison generation because people such as museum curators may now author an NLG system's knowledge base. The size of the user model is also flexible because the knowledge base author defines it.

The *content selection* component is almost entirely based off of ILEX. *Entity nodes* represent either museum objects or classes of museum objects. Lines connect these nodes to represent the relations between them. *Fact nodes* hold *predicate information* for each entity. Also like ILEX, *relation nodes* hold links between facts, including whether linked facts compare to each other or contrast each other. These nodes also follow Rhetorical Structure Theory by containing only one *nucleus* and its corresponding *satellites* [15]. The *content potential* is generated once all nodes are compounded into one graph [10].

System Architecture

The Exprimo architecture resembles that of a typical NLG system. However, the authors describe it as more modular in structure than ILEX [16]. The *domain model* represents Exprimo's knowledge base. Similar to ILEX, the *content selection* component generates a graph of information about museum objects from the knowledge base. To select facts to mention in an object description, Exprimo employs ILEX's parameters for *relevance* to the current object. These parameters include the perceived awareness and importance of facts by the user. Exprimo's *text planner* produces an abstract plan of generated text about the selected museum object. It also arranges the plan into a rhetorical structure modeled off of the *relation nodes* found in the content potential. Similar to ILEX, Exprimo only includes the relation nodes that correspond to facts mentioned in the text plan. However, Exprimo extends ILEX's planning process by incorporating a *microplanner*. The microplanner fleshes

out the first-cut plan produced by the text planner in preparation for surface realization. Since M-PIRO is a multilingual system, Exprimo selects one language-dependent grammar from the surface realization component to generate the output text.

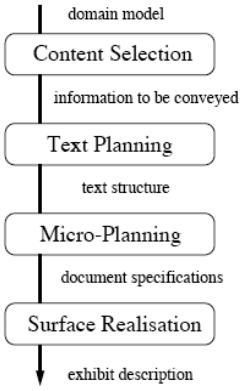

Figure 2.5: Exprimo system architecture [10]

Context Representation

M-PIRO's context representation is very similar to ILEX. Like ILEX, it is represented in three ways: (1) a *discourse history*, (2) *local and global focus*, and (3) a *user model*. The M-PIRO system treats all user-selected hyperlinks as a type of *discourse history*. Like the Migraine, PEBA-II, POWER, and ILEX systems, M-PIRO stores a record of the facts that have already been mentioned to avoid repeating information. M-PIRO also uses the discourse history to vary sentence structure and discuss previously mentioned objects. The discourse history is imperative in order for Exprimo to make comparisons between the currently viewed object and those previously mentioned. *Local* and *global foci* are implemented in the same way as ILEX.

The greatest difference between the context representations for Exprimo and ILEX lies in the *user modeling* component. The *user model*, similar to ILEX, holds weighted parameters about the perceived awareness (i.e., how likely the user is to be aware of the fact) and importance of facts. As an extension to the ILEX system, M-PIRO stores a model of each user on a remote server. Thus, unlike ILEX, users can revisit the M-PIRO system multiple times with their previous user models. Another improvement over ILEX is its ability to tailor information, including the complexity of comparisons, to people of different ages and expertise, like the PEBA-II system [4]. This permits the system to interpret each user model's weighted parameters differently depending on the user type. Unfortunately, it still does not follow a sound metric for the awareness and importance parameters for facts.

Potential for Applicability across Domains

Exprimo has not been shown to work robustly to generate text in domains beyond museum objects. However, the authoring tool should allow people to build knowledge bases in other domains. Knowledge base authors do not need to be technically proficient in handling the idiosyncrasies of past NLG systems' knowledge bases.

2.1.8 Conclusion

This review discusses a variety of NLG systems that generate comparisons based on stored contextual information. All attempts at storing contextual information relate directly to the present interaction with a user. We also assess each system's approach to generating comparisons from a knowledge base. Until recently, a rigid, system-defined algorithm could generate comparisons. This restricts the potential for tailoring comparisons based on user preferences. The Methodius system is the first to suggest that parameters set the type of comparisons an NLG system can generate, given the facts to be included in the comparison.

As we have seen, each NLG system's method for comparison generation suggested improvements through future research. The TEXT system could not generate comparisons unless explicitly asked for by the user. In the Migraine system, there are very few restrictions placed on the types of comparisons that can be made between medications. The PEBA-II system determined the importance of mentioning a similarity or difference by accessing an author-defined pre-ranked list of weighted attributes. Since POWER attempted to extract knowledge base information from a relational database, generated texts, including comparisons, were much more basic than those produced from the handwritten knowledge base. ILEX did not always include comparisons in its object descriptions. M-PIRO's process of fact selection for comparisons limited the number of potential comparisons where user preference scores could be more appropriate.

2.2 Evaluation of Comparisons in NLG Systems

To date, research has been conducted in the evaluation of systems that generate comparisons. Mellish and Dale divide the evaluation problem into three types. First, an evaluation of the theory behind a NLG system (e.g., *Rhetorical Structure Theory*) could be done. This evaluation could investigate whether a theory could be applied across domains and if it is suitable for the generation task [18]. In addition, another type of evaluation is that of the properties of a NLG system. For instance, this could be a comparison of two types of generation algorithms. In our study, we are evaluating Methodius' parameterized comparison generation algorithm. Lastly, we may want to evaluate a NLG system for its possible use as an application over other approaches (e.g., use of a human writer) [18]. As part of our evaluation of Methodius, we would like to assess its potential as a text generator of music facts in a "digital disc jockey" application.

An experimental evaluation of the ILEX (Intelligent Labeling Explorer) system, a predecessor of Methodius, found that text tailored to a user's browsing history of museum jewelry did not improve participants' scores on factual recall questions versus static text. This tailored text (i.e., dynamic hypertext) only included the ability to generate comparisons and to maintain a history of which facts the user has already read about [19]. The lack of the ability to aggregate multiple facts into sentences may have contributed to their surprising results.

Karasimos and Isard conducted an evaluation of comparisons and aggregations of multiple facts into sentences in the M-PIRO system. They used M-PIRO's knowledge base of ancient Greek artifacts for their study. They conducted an experiment where participants read two sets of texts, one about coins and one about vessels. One of the two sets contained comparisons between objects and aggregated multiple facts into sentences [8]. After reading each set of texts, participants answered a series of questions that assessed how well they remembered the facts presented in the texts. At the end of the experiment, they conducted a survey that asked participants to subjectively evaluate the texts generated by M-PIRO. The results of these two question types agreed with their hypotheses. They found that participants learned more and perceived that they learned more from texts that contained comparisons and aggregations over texts that did not. We hope to parallel these results in our study.

Chapter 3

The Methodius Natural Language Generation System

3.1 The Methodius NLG Domain and Architecture

The Methodius NLG system is descended from the Exprimo system, but was completely reimplemented to provide a more robust and scalable system, and the grammar component was changed from Systemic Functional Grammar to Combinatory Categorial Grammar. In addition, a more sophisticated comparison generation process was added. The authors claim that Methodius can be used to generate text for any domain where a database of objects and their respective attributes can be created [9]. This is because Methodius can be used with an authoring tool to create a knowledge base following the "object-attribute" ontology featured in previous systems dating back to the TEXT system [16]. Their initial system domains contained knowledge bases of Scottish monuments and Greek artifacts.

Comparison Generation

The Methodius NLG system forms customizable descriptions of objects from a defined database. Methodius features a novel algorithm for generating comparisons between objects that are currently being described and those that have previously been seen. This comparison-generating algorithm stands out from previous attempts because it chooses the most relevant and interesting comparisons given a context that is set by several explicit parameters [9]. These parameters set the degree of importance of the following factors:

A) Comparisons are preferred to be between larger groups as opposed to smaller ones. This factor depends on α, the number of objects in a group of previously mentioned objects to which the current object can be compared.

B) Comparisons are preferred to have as many facts to compare as possible. This factor depends on β, the number of possible comparisons that can be made between the current object and a group of comparable objects that was previously mentioned.

C) Comparisons are preferred to be between objects that are similar. This factor depends on γ, the number of edges on the taxonomical classification of objects between the current object and a group of comparable objects that was previously mentioned.

D) Comparisons are preferred to be between recently encountered objects. This factor depends on ϕ, the number of objects mentioned between the current object and the most recently mentioned object in a group of comparable objects.

E) Since the user can only remember a limited set of previously mentioned objects, the size of the group of comparable objects that were previously mentioned is also limited [9].

Factors A-D may be given weights to influence Methodius' approach to generating comparisons. The system assigns the weights to each parameter depending the significance of each factor. At the present stage of this research, the author suggests that weights for factors A-D should range from 0 to 1. The system first selects the set of possible groups of comparators to the current object [10]. Each member of the set holds a subset of grouped objects. For each group, Methodius uses information about the group to assign values to α, β, γ, and ϕ [9]. Each of these values is then multiplied by its *corresponding user-set weight* and summed as follows:

$$score = (\alpha \text{ X } memb) + (\beta \text{ X } comp) - (\gamma \text{ X } hierdistance) - (\phi \text{ X } histdistance)$$ [9]

Figure 3.6: Parameterized comparison generation algorithm formula, where α is the number of objects in a group of previously mentioned objects to which the current object can be compared;
β is the number of possible comparisons that can be made between the current object and a group of comparable objects that was previously mentioned;
γ is the number of edges on the taxonomical classification of objects between the current object and a group of comparable objects that were previously mentioned;
ϕ is the number of objects mentioned between the current object and the most recently mentioned object in a group of comparable objects.

Methodius then ranks all groups by their score in this formula. The group with the highest value is selected for comparison with the current object. If multiple groups tie for the highest score, then one group is chosen at random [9]. This suggests that additional constraints on comparison selection are needed.

Different types of comparisons may be generated depending on the properties of the selected group. A feature of the group is compared to the same feature in the current object if members of the group share the same feature (e.g., *time period of origin*). Methodius makes an *illustrative comparison* when the shared feature is similar to that of the current object, in a style similar to the PEBA-II system [4]. The system *contrasts* the group with the current object when the shared feature differs from that of the current object.

Knowledge Base Structure

Methodius' knowledge base can be built using M-PIRO's authoring tool. Thus, Methodius makes no changes to the structure of the knowledge base [10]. To take advantage of the M-PIRO authoring tool, the authors of Methodius developed a conversion script to convert M-PIRO knowledge base output into the proper input for Methodius.

3.2 The Methodius Pipeline (including OpenCCG)

Methodius adopts the Exprimo pipeline. As such, the improvement to comparison generation occurs during the *content selection* process [9]. Before running Methodius, the knowledge base built with the M-PIRO authoring tool is exported into a collection of .zip files containing a detailed ontology of the music domain, along with the songs, people, and other attributes that populate it. We then convert the knowledge base output from the M-PIRO authoring tool into the appropriate format for Methodius. Next, we export music domain information into a format suitable for OpenCCG, an open source library for the Combinatory Categorial Grammar formalism [20]. Methodius uses OpenCCG for its text generation component. Finally, upon execution at the command line, Methodius generates natural language text through the same pipeline as Exprimo. This includes the phases of *content selection*, *text planning*, *microplanning*, and *surface realization* mentioned in the previous chapter [9].

Context Representation

Although Methodius improves upon Exprimo, it represents context in the same manner. This includes components for tracking the *discourse history*, maintaining *local and global foci*, and storing a *user model* [10].

Potential for Applicability across Domains

The current test domain for the Methodius project is one composed of cultural artifacts [9]. The comparison algorithm featured in Methodius has not yet been proven to generalize across domains. Several upcoming projects will attempt to compose knowledge bases in new domains, such as music information and ancient Greek museum objects.

Future research should investigate if these parameters are sufficient for varying comparison generation across domains. We found Methodius' parameterized comparison generation algorithm to be flexible enough for our research in comparison generation from context.

Chapter 4

Obtaining Disc Jockey (DJ) Transcriptions

4.1 Method

To develop a user study to investigate our hypotheses, data must first have been collected to understand the type of facts disc jockeys tend to say about music. We focused on two genres where music descriptions between songs were common, jazz and classical music. We focused on classical music transcriptions because they were more common on the radio station we tuned into, BBC (British Broadcasting Corporation) Radio Three [21]. The radio streams were transcribed from several disc jockeys, including those from the shows "Composer of the Week", "Afternoon on 3", and "In Tune". All transcriptions were performed for personal non-commercial use. By distributing our transcriptions over several shows, we were able to acquire a greater sample space of disc jockey speech and their respective styles between songs.

We transcribed sixty-four examples of what disc jockeys discussed between songs. To maintain uniformity in our transcriptions, we followed the Linguistic Data Consortium's transcription guidelines [22]. The radio was played using Apple Inc.'s QuickTime 7 and transcribed into the open-source text editor Aquamacs [23, 24]. The transcription process was mostly straightforward. One challenge came when transcribing proper nouns into text. Fortunately, correct proper nouns were not crucial for this study. Advertisements and lengthy descriptions of music-based artists longer than one minute long were omitted from the transcriptions. Note that this was not a thorough corpus collection; the purpose of collecting examples was to gain a sense of what attributes of songs disc jockeys tend to discuss and compare.

4.2 Observations from Trends in DJ Transcriptions

By tuning into this public radio station, we found disc jockeys to have a variety of styles while on the air. Most disc jockeys tended to be rather verbose and varied in their sentence structure, using sentence generation strategies beyond the scope of the Methodius system. Often implicit comparisons were made to convey a theme between multiple consecutive music pieces. This strategy helps guide the listener through several music pieces with a sense of awareness of the variety of potential similarities between music pieces.

> "I will continue this exploration of the world of the Troubadours with an example of an *alba*, or dawn song, the Troubadour speciality."

Figure 4.7: Quote from sample disc jockey transcription.

Time periods were also crucial to provide the listener with a frame of reference for when (and possibly where) music pieces were made.

> "And here's a sequence of characteristic songs and instrumental pieces from that age of chivalry starting with a pairing performed by Musica Antigua."

Figure 4.8: Quote from sample disc jockey transcription

Facts about composers and performers were also frequently mentioned. These included where the artist was originally from.

> "The talented Troubadours sang and played their way to fame and fortune. Barenot da Vontadon was one of the most famous of all. He was born in the Limiza, the son of a servant who fired the ovens in the castle of Vontadon from which he took his name."

Figure 4.9: Quote from Sample Disc Jockey Transcription

Conductors, performers, and composers were occasionally mentioned together in the same sentence. Musical influences to the artist were also frequently mentioned.

> "[That was] Jones conducting the BBC Symphony Orchestra in 'Out of the Mist' by Lillian Elkington who studied with Bantock."

Figure 4.10: Quote from Sample Disc Jockey Transcription

Where appropriate, disc jockeys gave detailed information about the performers.

> "[This piece] emerged into this full maturity, his second piano quartet, played there by Domus; the pianist Susan Combs, violinist Krisis Oscostovich, the oboe player Robin Ireland and cellist Timothy Hugh."

Figure 4.11: Quote from Sample Disc Jockey Transcription

Disc jockeys naturally discussed facts about the music by aggregating facts together.

> "The symphony was played by the Philharmonic Orchestra conducted by Sir Andrew Davis in a concert for the Queen Elizabeth hall in London."

Figure 4.12: Quote from Sample Disc Jockey Transcription

The record label was also discussed by disc jockeys from time to time.

> "In 'By Berliers,' the Danish National Symphony Orchestra and choir, conducted by Thomas Gasda, [a] new release on the Chandoffs label."

Figure 4.13: Quote from Sample Disc Jockey Transcription

Disc jockeys tended to give their own subjective opinions on music pieces. We avoided this in our study because it would have added an unnecessary variable to our evaluation of comparison generation. This is primarily because a participant might disagree with the digital disc jockey, thereby producing an undesirable bias.

> "Wonderful music [there] from Georgia."

Figure 4.14: Quote from Sample Disc Jockey Transcription

From these transcriptions, we devised a strategy for planning out the text our digital DJ would say in our experiment. First, we considered any text mentioned by human disc jockeys to be regarded by people as interesting to some degree, and included that type of fact in our knowledge base. For example, for each music piece, we included information about the performing artist, the artist's influences, the time period of the music piece, the record label of the music piece, and the writer of the music piece. For jazz music, we prioritized mentioning the performer first, and the writer second. Instead, for classical music, we prioritized the composer of the music piece first, and the performer and conductor, if any, second. This was a common property in virtually all the disc jockey transcriptions. We also intended to keep our generation system as neutral as possible with regard to feelings toward specific music pieces.

Chapter 5

Knowledge Base Construction and Text Generation

5.1 Introduction

Following the transcription of disc jockeys' descriptions of music between songs, we carefully selected and handwrote twelve database entries of songs, the number required for our experiment. First, we developed the ontology of attributes of jazz songs and classical music pieces based on the most common attributes that DJ's mention. Next, we selected songs from the allmusic.com music database [25]. For each song, we entered the song and its associated attributes obtained from the allmusic.com database into our own knowledge base using the authoring tools developed by the M-PIRO project. This task was broken down into two types, "domain" authoring and "exhibit" authoring. We divided the task in this manner to follow the authoring process of the M-PIRO project [16].

5.2 Domain Authoring

During the "domain" authoring process, we developed a single-inheritance ontology for a knowledge base of music pieces. This first required us to list the high-level entity types in the music domain. These included entity types such as "song", "person", "instrument", "classical music period", and "jazz music period". A complete ontology is displayed below in the left panel of the authoring tool we used to build the knowledge base. We expanded the ontology of our domain based on the common attributes of music that were mentioned in our sixty-four disc jockey transcriptions. These included adding entity types about the active years of an artist ("artist-years-active"), the composer, the conductor, and the performer of a music piece. Information from the disc jockey transcriptions also motivated us to add entity types for the record label of a music piece, the album of a music piece, and the album's release date.

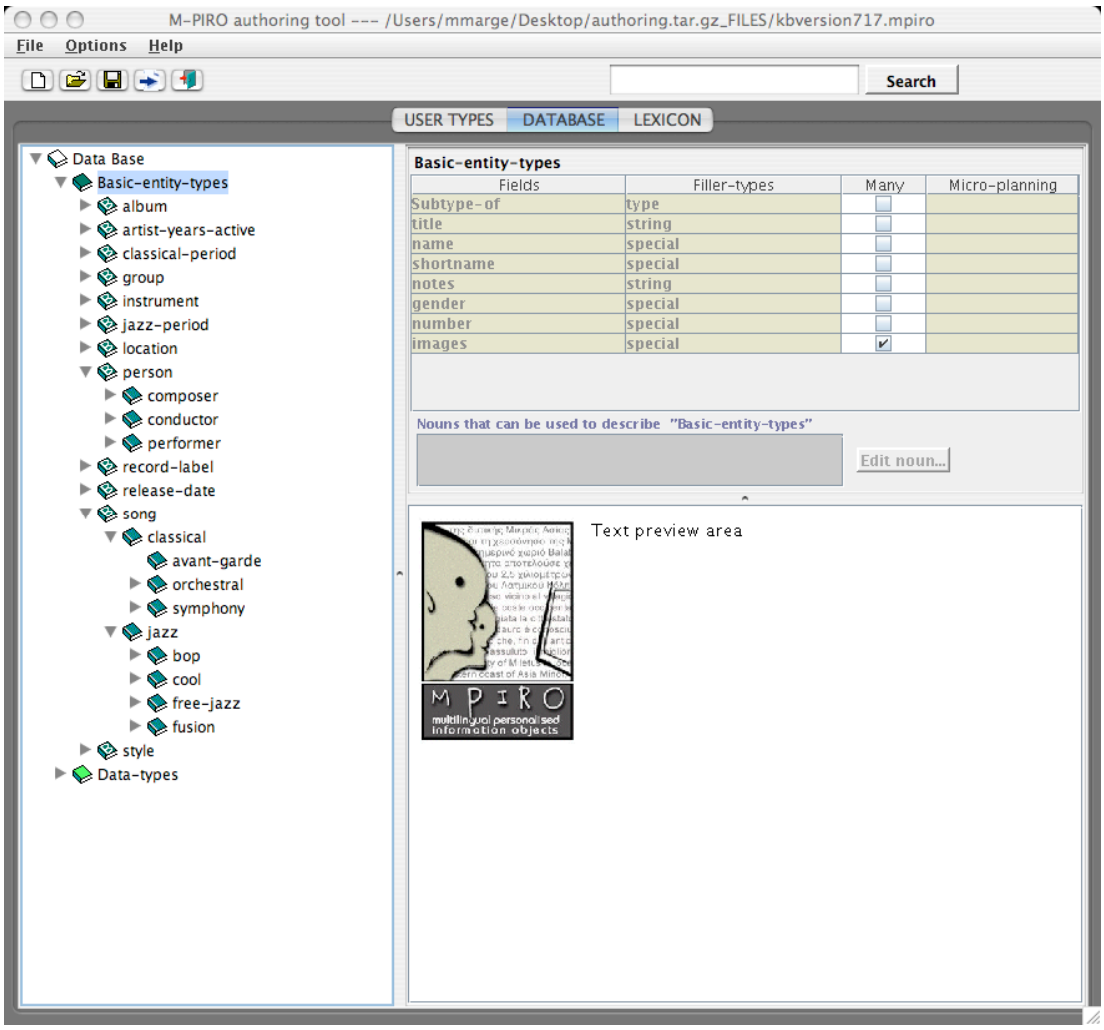

Figure 5.15: Ontology of the music domain (shown in the left panel).

We also developed a hierarchy of song types. The highest level of the hierarchy consisted of the song types of *jazz* and *classical*. The subtypes of each of these music genres were defined according to the allmusic.com database [25]. Further subtypes included attributes about the time period of the music. We also acquired these time periods from the allmusic.com database.

Using the M-PIRO domain authoring tool, we defined "fields" (or *relationships*) for each entity type, which express relationships between entities, such a relationship between music pieces and music periods. For example, the *classical-period* field must contain a string (i.e., a *filler*) describing a classical music piece's time period, which is of *filler* type "classical period". Every *filler* must have a specified entity type in order for an entity of that type to "fill in" the *field*. For our study, only many-to-one or one-to-one relationships between *fields* and *fillers* were necessary. For instance, songs may only come from one music period in our study. In each field, we also specified its "microplanning expressions," which provide detail on how the field's expressed relationship should be discussed at the sentence level [16]. We specifically built microplanning expressions that served as plans of which verbs, prepositions, and modifiers to generate when a particular field is selected during user interaction. In most microplans, a verb for a relation is specified, such as the verb *compose* for the relation *composed-by* for a song. Then, the voice and tense of the verb is specified. Most fields used the passive voice of a verb in the past tense. Lastly, a preposition, where appropriate, is specified to connect the field's entity type (in this case, *song*) with a corresponding entity type (in this case, *composer*). An example of a set of fields is shown on the following page for the "classical" entity type.

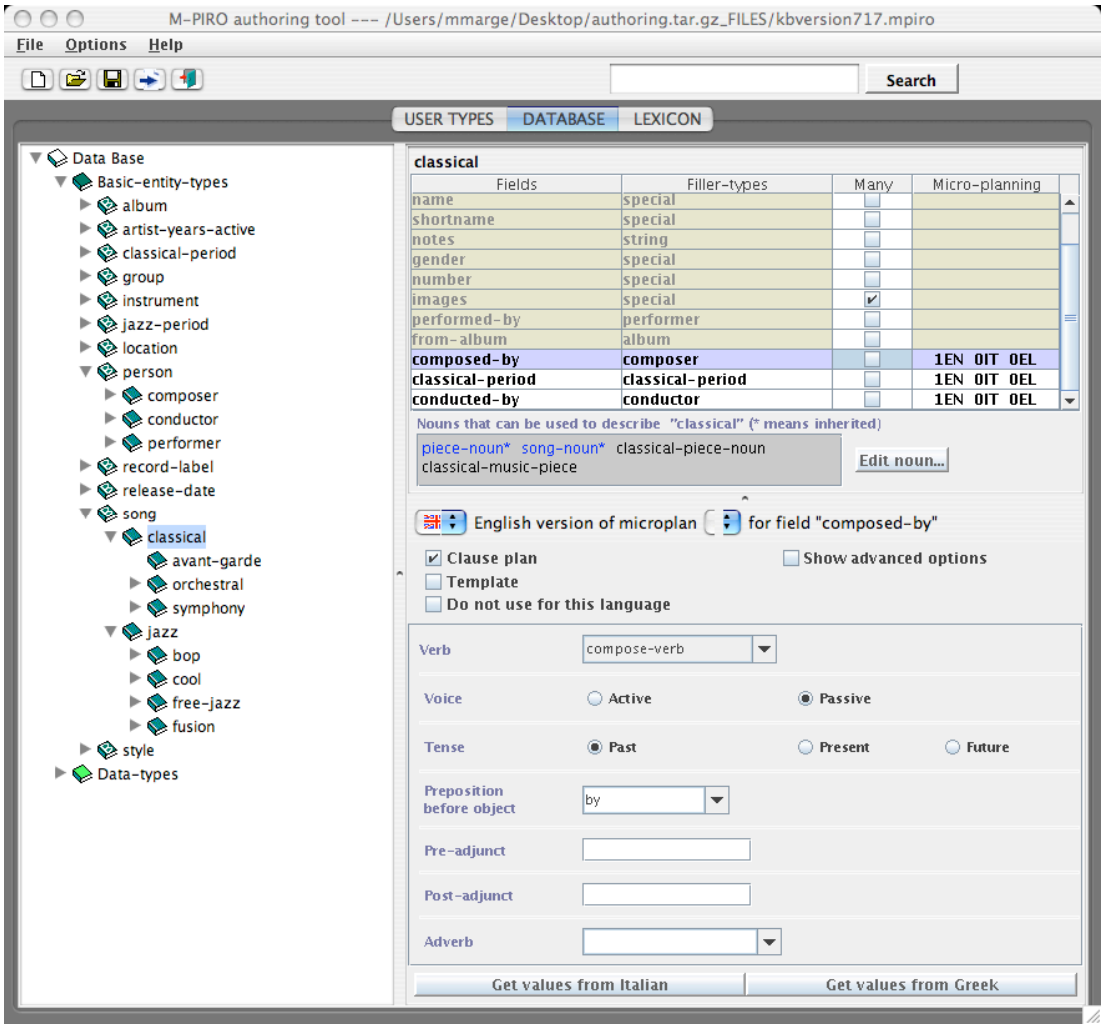

Figure 5.16: Fields for the "classical" entity type is shown in the upper right panel. A "microplan" is shown for the *composed-by* field in the lower right panel.

The microplan for *composed-by* specifies "[*song*] *composed by* [*composer*]". Entities are then added in the form of individual songs and composers during the "exhibit authoring" process. Note the Methodius system does not use the M-PIRO microplanning features "pre-adjunct", "post-adjunct", and "adverb" because they were not required for this study.

We then added all the necessary lexical entries (i.e., nouns and verbs) dependent on the music domain into the knowledge base. Every noun is associated with at least one entity type at the appropriate level of the ontology. Note that for Methodius, nouns entered into the lexicon are essentially noun phrases without determiners, where determiners are added later during generation. To vary sentence structure during generation, as many synonyms of the entity type as possible were added. For example, in the figure above, the *classical* entity type first inherits the nouns "song" and "piece" from the *song* entity type. Next, we added the nouns "classical piece" and "classical music piece" to the list of nouns that could describe the *classical* entity type.

Verbs may be used in multiple microplans. For example, for the *jazz* entity type, the verb *write* is used for *written-by* and *written-during-jazz-period*. A partial list of the nouns and verbs in the lexicon is shown below.

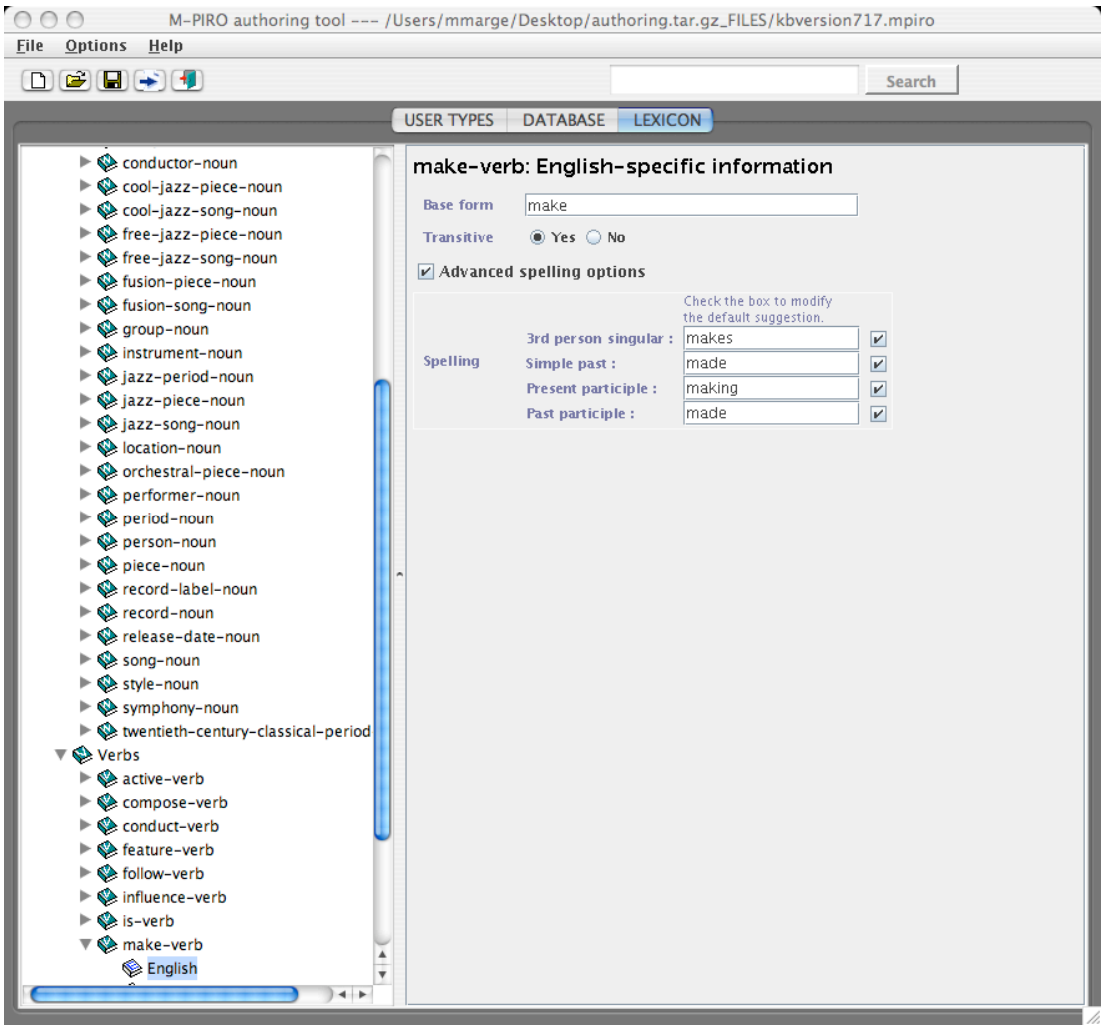

Figure 5.17: Nouns and verbs in the music domain lexicon are shown in the left panel. The right panel here specifies the verb *to make*'s text and tenses.

5.3 Exhibit Authoring

During the process of "exhibit authoring," we defined the twelve entries of songs necessary for our study. In total, there were six music pieces added to types of classical music, and six music pieces added to types of jazz music. These songs were carefully selected from the allmusic.com database to yield at least two interesting comparisons when placed into a specific order [25]. We were able to ascertain potential comparisons by looking for similarities between the fields of entities. For example, two music pieces we used in our study, "Adagietto" and "Molto Moderato", were from the *Romantic* classical music period. This meant that when Methodius searched for potential comparisons, a statement pointing out this similarity could be generated. Each song entry also had all of its other fields filled with appropriate related information.

In addition to adding song entities to our knowledge base, we populated all other leaves (i.e., entity types at their deepest levels) of our music ontology. This required us to add all the necessary entities for all other entity types that were tied to the twelve songs. For example, each song's album, performer, and composer were added to their corresponding entity types. For each album, its record label and release date were also added to the knowledge base. The locations of origin and musical influences of performers and composers were also added to the knowledge base. We added as much information about songs as possible from the allmusic.com database to our knowledge base that fit properly into our ontology. This is because we wanted our generated text from Methodius to seem natural and varied to participants in our experiment.

Once this phase of authoring was complete, we had a knowledge base in an XML format appropriate for an existing grammar formalism, Open Combinatory Categorial Grammar (OpenCCG) [26]. This required us to spend time becoming familiar with OpenCCG's architecture and how facts are embodied in the OpenCCG format. A research fellow helped us become familiar with OpenCCG.

5.4 The Methodius/M-PIRO Authoring Tool

Presently, there are no existing knowledge base authoring tools exclusively for Methodius. To approach this problem, we instead used the existing M-PIRO authoring tool. The tool was made compatible with Methodius by converting the knowledge base output of M-PIRO into a format suitable for Methodius. Although the M-PIRO interface features the ability to build knowledge bases in Italian, Greek, and English, we only employed the English toolkit for our experiment.

The M-PIRO authoring tool is a Java-based, universally-compatible graphical user interface for the processes of *domain* and *exhibit* authoring [16]. It took approximately one day to learn all the features of the interface, a low learning curve compared to other knowledge base construction methods [27]. There are three high-level features of the M-PIRO interface: defining one or more "user types", designing a "database" (i.e., a knowledge base) of facts for use by Methodius, and declaring all nouns and verbs used by Methodius in a "lexicon".

5.4.1 User Types

The M-PIRO authoring tool features the ability to define multiple user types for a generation system, such as for experts and novices of a domain. A user type defines the number of facts that should be mentioned in a generated sentence. For the purposes of our experiment, we created only one user type, *adult*. To keep experimental conditions constant across our experiment and that performed by Karasimos and Isard, we set the maximum number of facts per sentence at four [8].

5.4.2 Knowledge Base

The tool allows for an ontology of *entity types* to be created. In our case, these entity types included objects and abstract concepts such as "song", "performer", and "composer". In the M-PIRO interface, entity types are added in the left panel under the "DATABASE" tab. The process of adding entity types to the knowledge base resembles the construction of a tree, with the root being the node "Basic-entity-types", which serves as a placeholder. In order to add sub-entity types that inherit the properties of an existing entity type, such as *classical* and *jazz* for the entity type *song*, one simply inserts a new sub-entity type into *song*. These added entity types inherit all properties of the entity types above them in that particular branch of the ontology tree.

All entity types must also be specified by at least one noun. In the M-PIRO interface, nouns for entity types are specified in the small gray box in the center of the right panel. This requires the desired entity type to first be selected. Nouns, defined in the lexicon, permit Methodius to refer to an entity type in a generated sentence. In the figure below, the nouns "song" and "piece" refer to the *song* entity type. These nouns will also be inherited by any of the entity type's children.

Relationships (or "fields") between entity types can then be added to any entity type as needed. For example, in our domain, these included such relations as "song-performed-by [performer]" for the *song* entity type and "performer-played-the [instrument]" for the *performer* entity type. An entity type's relationships are inherited by any of its children. In the figure below, the classical and jazz entity types, along with all of their children, inherit the "performed-by" and "from-album" relationships. Microplans for each relationship were specified by selecting the corresponding entity type in the left panel, and then the appropriate relationship in the left panel. In the figure below, the *song* entity type and its corresponding "performed-by" relationship are selected.

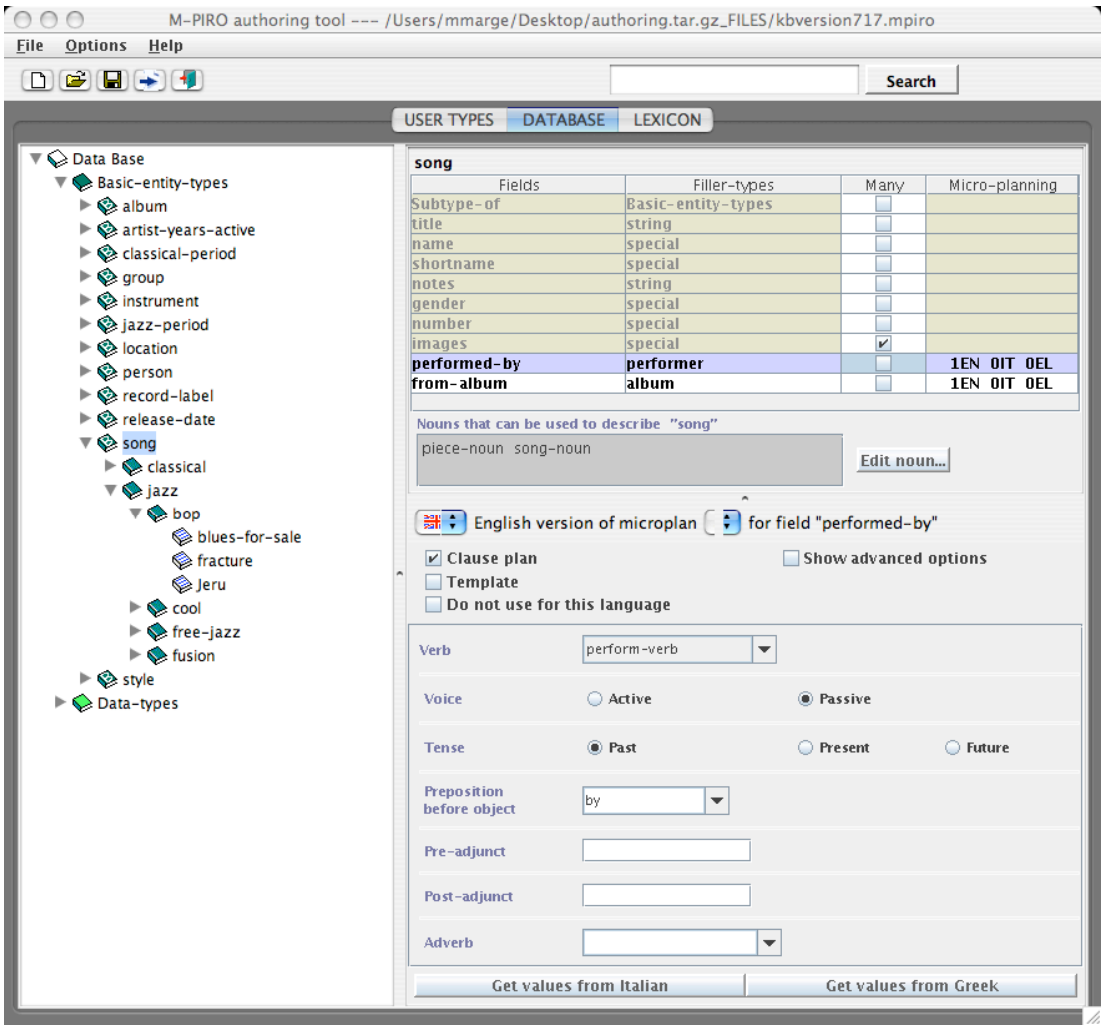

Figure 5.18: A detailed view of entity type *song*'s relationships. The microplan for *song's* "performed-by" relationship is specified in the lower right panel.

Given a specified ontology, the process of populating the knowledge base with entities for our domain was straightforward. This process is known as "exhibit authoring". For example, for each song we added to the knowledge base, we first identified its music type, as far detailed as possible. All songs in our knowledge base are specified as either "jazz" or "classical", along with a style specified by the allmusic.com database [25]. To add a song entity to our knowledge base, we first selected the most detailed entity type that describes it. Next, we inserted the entity as a direct child of that entity type. In the image above, for example, the song entity "Fracture" is specified under the entity type *bop*, which is a type of *jazz*. All other entities in our knowledge base were specified in the same manner, directly under the entity type that best described it.

Once the entity is added to the left panel ontology, we may then provide details for the selected entity. There were two sets of details to specify for each entity. First, as in the image below, we entered the string of text that would describe the entity during text generation. We also supplied the gender of the entity, among the choices of “neutral”, “masculine”, or “feminine”. Then, we specified the plurality of the entity as either “singular” or “plural”. This information assisted Methodius by specifying the details necessary for following proper rules of grammar in English. The figure below shows these details in the M-PIRO authoring tool.

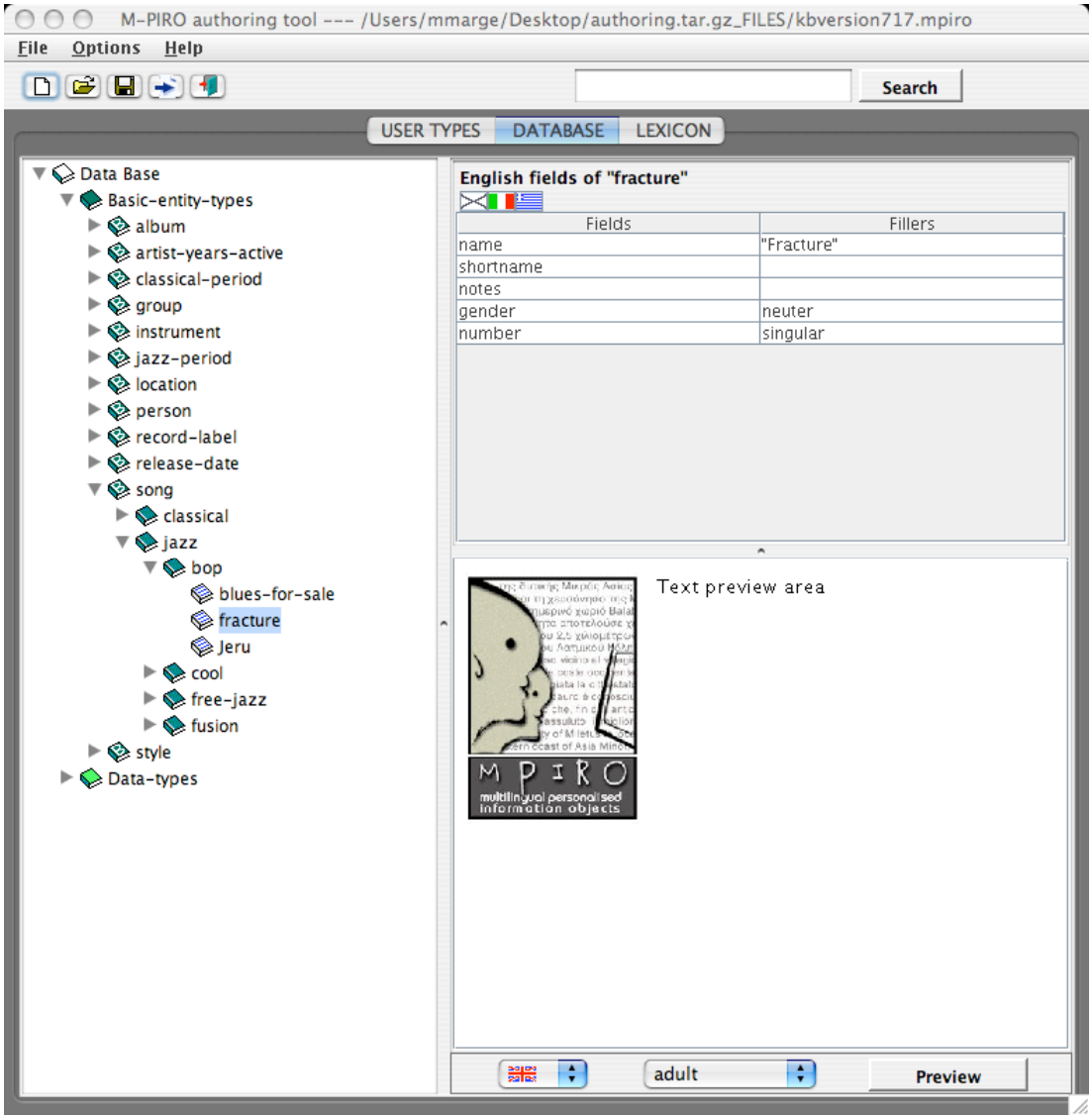

Figure 5.19: A detailed view of entity *Fracture*’s first menu of detail, specifying properties of the string describing the entity.

The second menu of detail for an entity allows the user to specify the *fillers* for all relationships inherited by the entity. For our experiment, we defined as many entity relationships as possible to permit Methodius to vary the facts it mentions in generated sentences. In our example for the entity *Fracture*, we specified the song's performer, album, writer, and time period by selecting the appropriate corresponding entities in the knowledge base. These entities must also have been defined in order to be included in a song's relationships. We took a bottom-up approach to defining entities in our knowledge base, where the entities that had the fewest relationships were declared first. These entities included locations of origin, such as those for performers and instruments. Thus, entities with a large number of relationships, such as songs, would have all of their corresponding entities (e.g., albums, writers, and performers) for relationships already available as *fillers*. The figure below shows the second menu of detail for an entity.

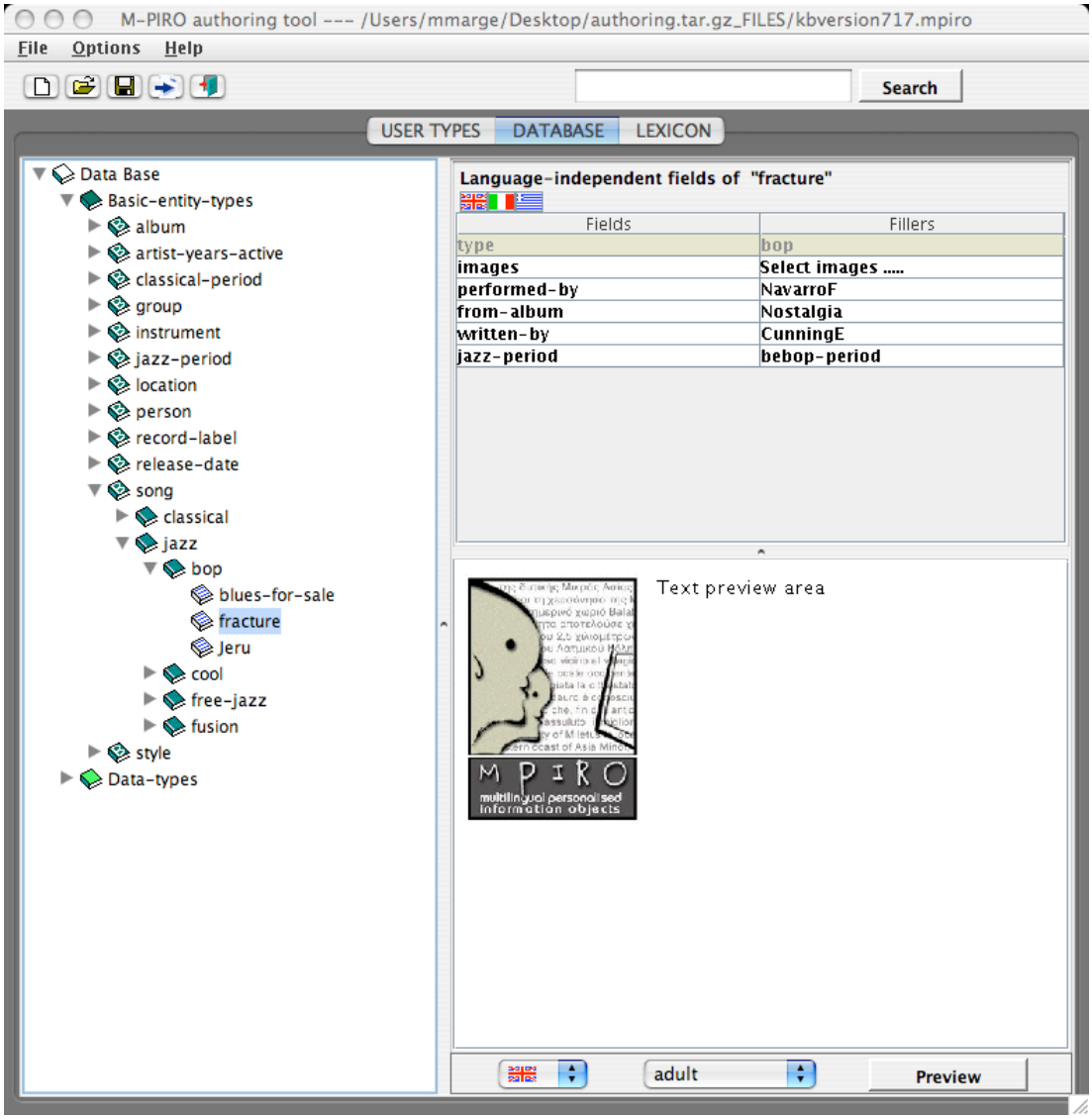

Figure 5.20: A detailed view of entity *Fracture*'s second menu of detail, specifying relationships.

5.4.3 Lexicon

All non-proper nouns and verbs that the system may generate were specified through the M-PIRO authoring tool's domain-dependent lexicon feature. The referring expression generation component in Methodius requires all entity types in the knowledge base to correspond with nouns in the lexicon [27]. As previously mentioned, nouns entered with the M-PIRO authoring tool are essentially noun phrases without determiners. Determiners are added later by Methodius. Every entity type must correspond with at least one noun. Each noun in the lexicon must be defined by its "base form", a string of characters that represent the noun. We also specified whether or not each noun is countable in the interface.

By default, Methodius appends an "s" to the base form to represent a noun in its plural form. If this rule does not apply to a noun, such as for the noun "symphony", the M-PIRO authoring tool permits us to directly enter the correct spelling of the plural form. In this case, we enter "symphonies" as the plural form. In the interface, we must tick the box to the right of the corrected plural form to verify that Methodius will use the corrected spelling. This feature is shown in the figure below.

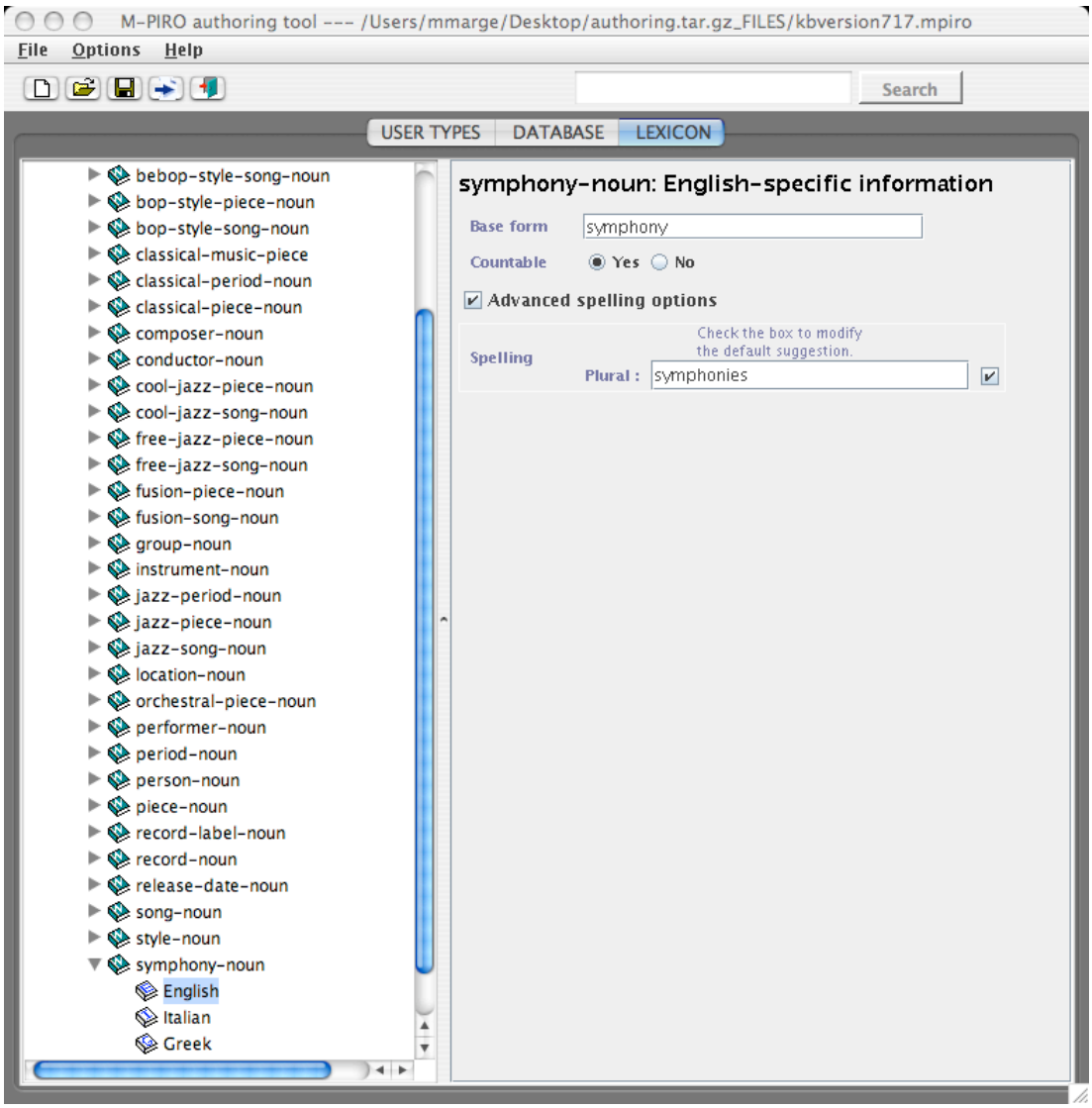

Figure 5.21: The corrected plural form of "symphonies" is displayed in the right panel of the M-PIRO authoring tool.

For our experiment, all verbs to be generated must also be specified in the M-PIRO interface. For each verb, we first entered the string of characters that represented the "base form" of the verb. We then specified whether or not the verb was transitive. By default, verbs whose base form ended in "e" had a *simple past* and *past participle* form that simply appended "d" to the base form. All other verbs were appended with "ed" in their *simple past* and *past participle* forms by default. The M-PIRO interface permitted us to specify these forms for irregular verbs, as shown in the figure below. These tenses were also confirmed for Methodius by ticking the boxes directly adjacent to the corrected spelling.

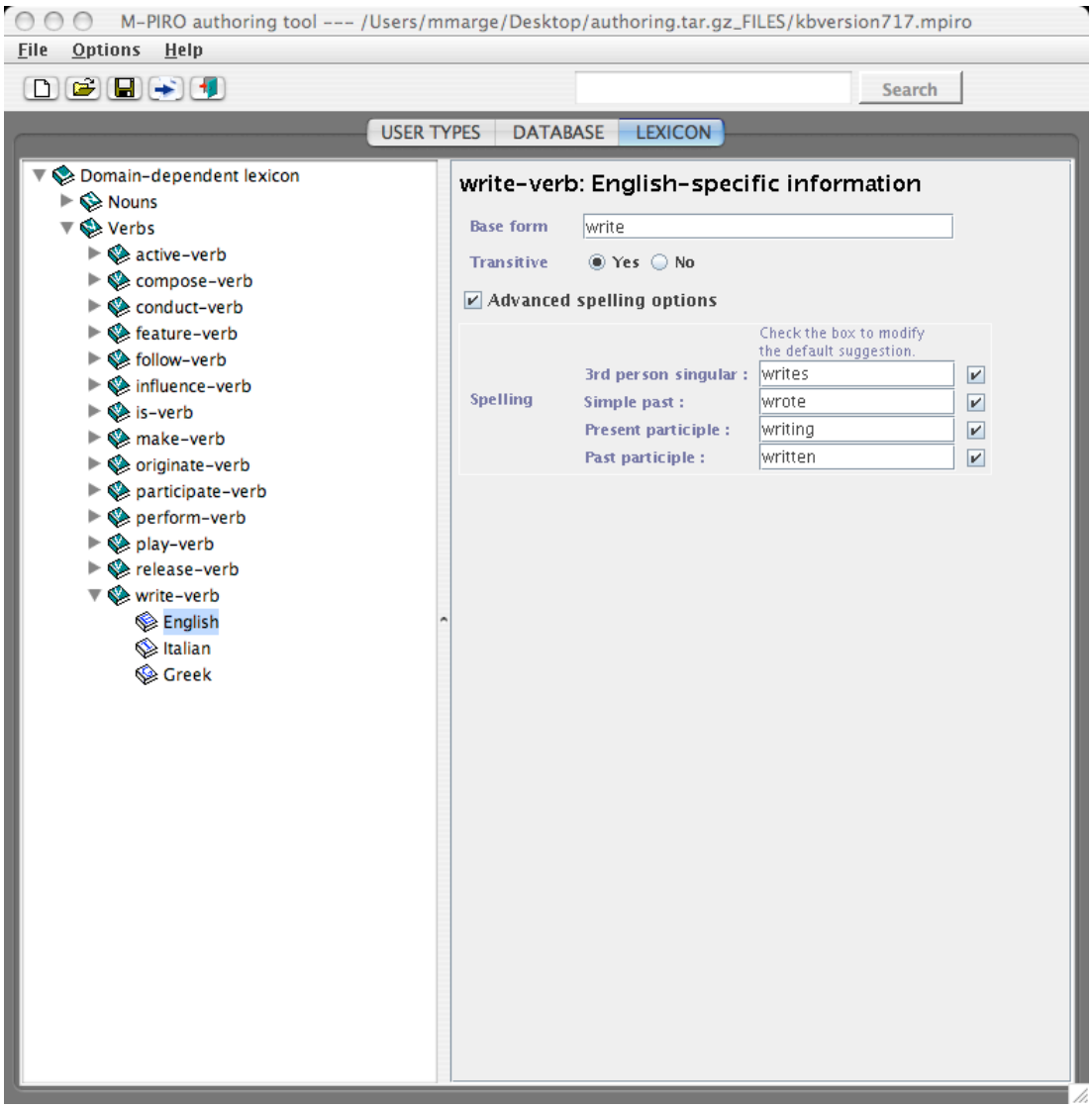

Figure 5.22: Verb specification for the irregular verb “to write”.

5.5 Executing Methodius

Once the knowledge base and lexicon were entered into the M-PIRO authoring tool, we exported and converted it into input appropriate for Methodius. Methodius first required us to load the domain of the knowledge base in order to overwrite any previous representations of the knowledge base in memory. Next, we modified a few entity type "comparison scores". These scores are used by Methodius' parameterized comparison generation algorithm to prioritize which entity types, if there are multiple choices, should be mentioned via comparisons first. For jazz music, we increased the *performer* entity type's comparison score so that it was mentioned first over the *composer* entity type. In our observations of jazz music disc jockeys, we found the performer to be mentioned first over the composer, if he or she was a different person than the performer. A composer may not be mentioned at all. However, for classical music, we increased the *composer* entity type's comparison score so that it was mentioned first over the *performer* entity type. This was because our classical music disc jockey transcriptions most often mentioned the composer first over the performer. However, unlike jazz music both are always mentioned for classical music.

For our experiment, we executed Methodius four times from the command line. Each execution of Methodius required us to first select any number of entities from our knowledge base, along with the number of facts to mention for each entity. We selected only six songs per execution for our study, with nine facts to mention per song. The resulting output was six paragraphs, one about each song. Each paragraph contained nine facts about the corresponding song in a series of three to five sentences. In the first execution of Methodius, we selected the six jazz songs from our knowledge base in a specific order that maximized the number of possible comparisons and we ensured that comparison generation was set to "active". In the next execution, we used the same six jazz songs in the same order, but set comparison generation to "inactive". We then executed Methodius for classical music, with six specific music pieces. We used the same six music pieces in the same order whether or not Methodius was set to generate comparisons.

Chapter 6

Pilot Experiment

6.1 Introduction

Our pilot experiment allowed us to test all aspects of our experiment before running our primary experiment. We wanted to verify that our experiment's instructions were easily understandable to participants. For our pilot study, we developed and tested our experiment on a remote server using the WebExp2 Experiment Design software [28]. We were also able to test our experiment's settings for variables such as the size of generated text and the number of facts per paragraph.

Our primary aim with our experiment design was to maintain as many conditions from a previous, similar study by Karasimos and Isard as possible [8]. This was because it allowed us to directly compare our results to their study's results. In their study, they set out to evaluate the M-PIRO text generation system's ability to both generate comparisons and aggregate multiple related facts together into the same sentence. They verified that people learned more and perceived that they learned more from text enriched with comparisons and aggregations of facts versus text that did not contain comparisons and aggregations of facts. Our experimental design is similar to theirs, however we permit all conditions of our experiment to contain text generated with aggregations of facts. Our aim is to *strictly* evaluate whether the presence of comparison generation alone will result in text that people will learn more and perceive that they learn more from than text without comparisons.

6.2 Method

6.2.1 Designing and Selecting the Song Texts

We used the allmusic.com database of music to carefully select a total of twelve songs for our experiment. We chose to have two types of music in order to maintain consistency with the Karasimos and Isard study, which also had two types of text. In their study, the domain of their knowledge base was ancient Greek artifacts. They presented participants in their experiment with six texts about coins and six texts about vessels [8]. We chose to maintain consistency with their text counts by adding information about six jazz music pieces and six classical music pieces so that we may be able to compare our results to theirs. In addition, having two sets of texts from different topics allowed us to test different conditions of our experiment with each text. For example, with two types of text (jazz and classical music), we are able to have a participant encounter a text set with comparisons, and a text set without comparisons. This yielded us a within-subjects design.

An experimental evaluation of the ILEX (Intelligent Labeling Explorer) system, a predecessor of Methodius, found that text tailored to a user's browsing history did not improve participants' factual recall tests versus static text. This tailored text (i.e., dynamic hypertext) only included the ability to generate comparisons and maintain a history of which facts the user has already read about [19]. We suggest that the lack of the ability to aggregate multiple facts into sentences may have contributed to their surprising results. Thus, we argue that our study is novel and will enable us to investigate whether comparison generation in a more modern system featuring aggregation will influence participants' ability to recall facts they were presented. In addition, our study investigates whether people perceive that they learn more from text with comparisons versus text that does not contain comparisons.

One challenge inherent in selecting these entities from a publicly available database was to eliminate as much common knowledge about the classical and jazz music pieces as possible. In order to decrease background knowledge as a potential factor in our experiment, we selected songs that primarily did not contain popular music pieces, performers, composers, and conductors. We were able to gauge the popularity of artists by their "popularity rank" in the allmusic.com database [25]. However, we had to maintain a careful balance between obscure artists and the ability to generate interesting comparisons. Obscure artists had less detailed information in the allmusic.com database as compared to popular music artists. For that reason, we were forced to select a few popular music artists for our experiment, as their music pieces had multiple possible interesting comparisons, a desired feature for our experiment.

We also had to decide on the types of comparisons that could be made in the texts. First, we listed the types of potential comparisons that each text type could make. For jazz text, comparisons between songs involving performers, albums, composers, and time periods were possible. Classical text could produce all four of these types of comparisons. Unlike jazz text, conductors could also be part of a comparison in texts about classical music pieces. Although the potential similarities for classical and jazz texts were not equal, we decided to include the conductor as a potential comparison for classical music. This is because across both text types, we will maintain the same number of generated comparisons for each text type. We limit Methodius to generating only five comparisons or contrasts per six paragraphs of text. Below is an example of a paragraph of text generated by Methodius with and without comparisons.

"The Great" with comparisons

Like "Molto Moderato", "The Great" was written during the Romantic period and it was composed by Franz Schubert. It was performed by the Royal Cambridgeshire Orchestra and it was conducted by Nikolaus Harnoncourt, who was active during the late 20th century. Nikolaus Harnoncourt originated from Berlin, Germany. "The Great" was from the album "The Symphonies", which was released on the Teldec label. The album "The Symphonies" was originally recorded in 1993.

"The Great" without comparisons

"The Great" was composed by Franz Schubert and it was performed by the Royal Cambridgeshire Orchestra. It was written during the Romantic period and it was conducted by Nikolaus Harnoncourt, who was active during the late 20th century. Nikolaus Harnoncourt originated from Berlin, Germany. "The Great" was from the album "The Symphonies", which was released on the Teldec label. The album "The Symphonies" was originally recorded in 1993.

Figure 6.23: A full entry for the music piece "The Great" generated by Methodius with and without comparisons.

As mentioned previously, we decided on the types of facts Methodius could generate for each music piece based on our disc jockey transcriptions. For any given music piece, Methodius could generate text about the song's performer, composer, album, recording label, and time period. Methodius could also generate text about a classical music piece's conductor. Text about a person's location of origin, the active years of the person, and the person's influences could also be generated. In addition, a performer's instrument and group, if any, could be mentioned. For jazz music, we decided to use the noun "writer" for the *composer* entity type, as the noun "composer" is used more exclusively for classical music. For our pilot experiment, we set the maximum number of facts per music piece to fifteen.

Like the Karasimos and Isard experiment, there is only one user type for our experiment, the *adult* type. Thus, Methodius can aggregate a maximum of four facts per a sentence in a generated paragraph. We kept the user type consistent in our experiment so that we may compare our experimental results with those of the M-PIRO study.

6.2.2 Designing the Evaluation Questions

We decided upon two types of questions to ask participants in this experiment. The first type of question, *factual recall* questions, was used to evaluate our main hypothesis that people learned more from text featuring comparisons and aggregations of facts versus text with only aggregations of facts. Similarly to the Karasimos and Isard study, after reading a set of six paragraphs about music pieces of a certain type, participants are presented with fifteen *factual recall* multiple-choice questions about the facts presented in the previous six paragraphs [8]. After answering all of the multiple-choice questions, participants are then presented with six more paragraphs about music and fifteen more *factual recall* multiple-choice questions about the facts presented in these texts.

Like the Karasimos and Isard study, only a portion of each *factual recall* question set was used to assess whether or not participants learned more from text generated with comparisons versus text generated without comparisons. We decided that **seven** multiple-choice questions of each fifteen-question set of *factual recall* questions would ask questions about facts that may be reinforced by comparisons. We call these "COMPARISON QUESTION" questions. The figure below presents two examples of multiple-choice questions that assess our hypothesis that people learned more from text with comparisons. The remaining eight multiple-choice questions in each section served as a control for this experiment. All potential letters were randomly assigned to each of the five potential answer slots (a through e).

Which period were "Molto Moderato" and "The Great" from?
a) the Postmodern Classical period
b) the Classic period
c) the Baroque period
d) the Romantic period
e) the Impressionist period

Which songs were performed by Fats Navarro?
a) "Alarm" and "Django"
b) "Fracture" and "Django"
c) "Avatar" and "Alarm"
d) "Fracture" and "A Mystery in Town"
e) "Avatar" and "A Mystery in Town"

Figure 6.24: Two examples of multiple-choice questions that assess factual recall from text that could be enriched by comparisons.

We intended to balance the types of factual information asked across both music types of classical and jazz. To ensure that an approximately consistent number of fact types were assessed in *factual recall* questions, we performed a tally. This tally is detailed in the table below. We were motivated to spread out the types of facts we asked about because we wanted to ensure our participants would not notice that we were explicitly asking only certain types of facts for the seven questions assessing comparison generation. This also prevented participants from "training" themselves to only remember certain types of facts from the paragraphs they read about. This also increased the difficulty of the *factual recall* assessment. One principal complaint of the Karasimos and Isard study was that the *factual recall* questions were too easy. We wanted to ensure that this would not happen in our study. To further increase the difficulty of the *factual recall* assessment, each question had five multiple-choice questions, one more than the Karasimos and Isard study. This decreased the rate at which random answers could influence the study.

Fact Type Queried	Classical	Jazz
Which album is M from?	2	3
Which period was M written during?	2	3
Where did P originate from?	1	1
What instrument did P play?	0	1
Which music pieces were performed by P?	2	2
Who was followed by P?	0	1
Which decade or time period was P active during?	1	1
Who wrote/composed M?	2	1
Which record label released A?	1	1
Who served as an influence to P?	1	1
Who conducted M?	2	0
Which pieces were conducted by P?	1	0

Table 6.1: A tally of the types of *factual recall* questions asked for our pilot experiment. **Legend:** M = music piece, P = person, A = album.

The second type of question, *post-experimental survey* questions, was used to assess participants' subjective perceptions of the generated text. We presented participants with twelve Likert-scale questions about their perceptions and beliefs at the end of the experiment [29]. In addition, we asked an open-ended question about their overall preference for either the classical or jazz music text and their justification for their position. We designed our questions based on the post-experimental survey presented in the Karasimos and Isard study [8]. Below are two examples of the questions we asked during this part of the study.

I learned a great deal from the text about jazz music.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I learned a great deal from the text about classical music.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

Figure 6.25: Two *post-experimental survey* questions. They are based on the Likert scale [29].

6.2.3 Designing the Web Experiment with WebExp2

In order to perform an appropriate user study, we developed a web interface that contained text generated by Methodius from our music fact knowledge base. We used the WebExp2 Experiment Design software to code all of the requirements of our experiment [28]. In our experiment, participants were asked to go through 5 stages:

(1) Read 6 paragraphs about a type of music (either jazz or classical).
(2) Answer 15 *factual recall* multiple-choice questions about those texts.
(3) Read 6 more paragraphs about the other type of music.
(4) Answer 15 more *factual recall* multiple-choice questions about those texts in Stage 3.
(5) Answer 12 *post-experimental survey* Likert Scale (1 to 5) questions about their subjective opinion of the texts (this section is constant across all conditions).

Within our WebExp2 directory, we developed four variations of our experiment, one for each experimental condition. Participants were assigned to one of four conditions:

Condition 1: Read about jazz music with comparisons throughout the texts (e.g., "Like "Molto Moderato", "Adagietto" is from the Bebop period.") *first* and classical music without comparisons *second*.
Condition 2: Read about classical music with comparisons throughout the texts *first* and jazz music without comparisons *second*.
Condition 3: Read about jazz music without comparisons *first* and classical music with comparisons throughout the texts *second*.
Condition 4: Read about classical music without comparisons *first* and jazz music with comparisons throughout the texts *second*.

Figure 6.26: Experimental design conditions.

The multiple choice questions don't change given the condition; so every participant saw the same two sets of 15 multiple-choice questions (varying order based on which music type comes first).

At first, we wanted to present participants with a 30-second clip of a song, as part of an experience of interacting with a "Digital DJ". However, we concluded that introducing this variable into our study would not prove helpful, and could have complicated our results. This is because music may influence a participant's emotional state, and it may provoke them into answering questions differently compared to other participants [30].

Instead, the interface was designed to present a paragraph of text generated by Methodius relating to the current song. This text may or may not have had comparisons. Once a participant finished reading the paragraph, he or she may then proceed to the next paragraph by pressing the "Next song" or "Next piece" button, depending on whether the music type was jazz or classical music, respectively. A figure indicating the participant's environment during this stage of the experiment is shown below. We decided that text paragraphs should be presented to participants as images for two reasons. First, we used images of text to keep the presentation of stimuli consistent across the array of computers that would be running the experiment. Secondly, this prevents the text from being selected by the participant, thus discouraging them from copying the text and placing it into another window as a reference to answer the *factual recall* questions asked later.

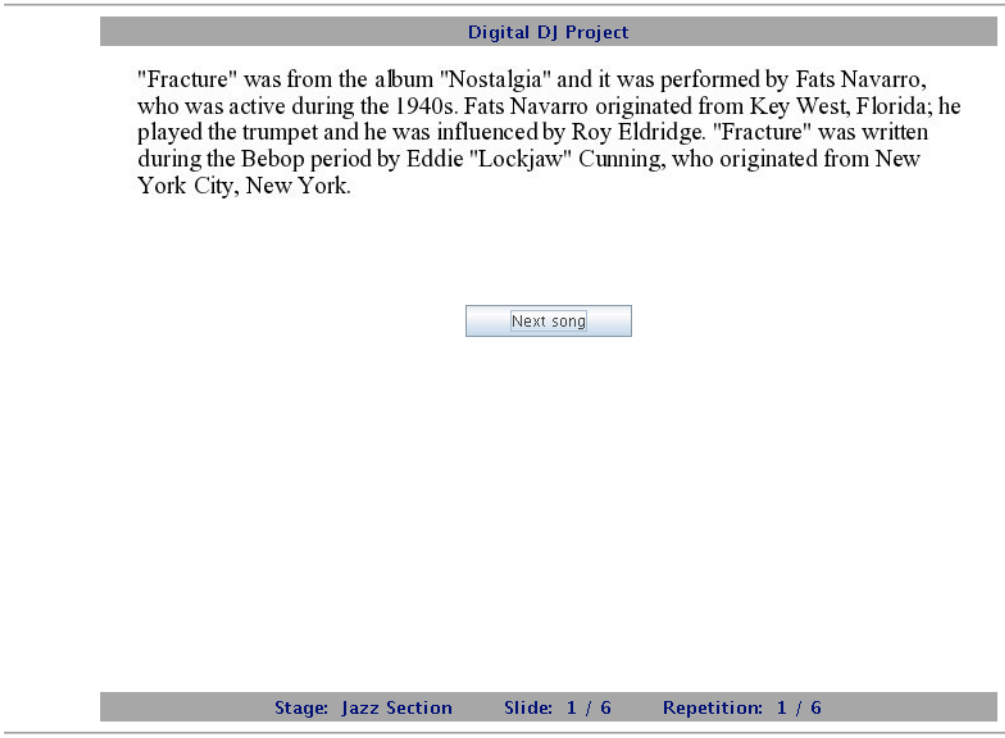

Figure 6.27: Participant's environment within a standard web browser during the portion of our experiment where generated text is presented in paragraphs.

Once participants finished reading six paragraphs about a type of music, they were presented with a second type of webpage. This type of webpage presented the participant with a *factual recall* question as an image. Like for other types of text presentation in WebExp, we used images for text to maintain the appearance of stimuli across computers. This type of webpage automatically focused the cursor inside a textbox where the participant must enter one of the five potential answers to the question. We limited the input for this textbox to either "a", "b", "c", "d", or "e". This prevented participants from making input errors during the experiment, and ensured that no data would be discarded due to an out-of-range answer. The WebExp interface also prevented participants from proceeding through the current question until an answer was selected, discouraging participants from rushing through the experiment. Once the participant entered his or her answer for the current *factual recall* question, they could proceed to the next question by pressing the interface's "Next question" button. A figure illustrating the participant's environment during this stage of the experiment is shown below.

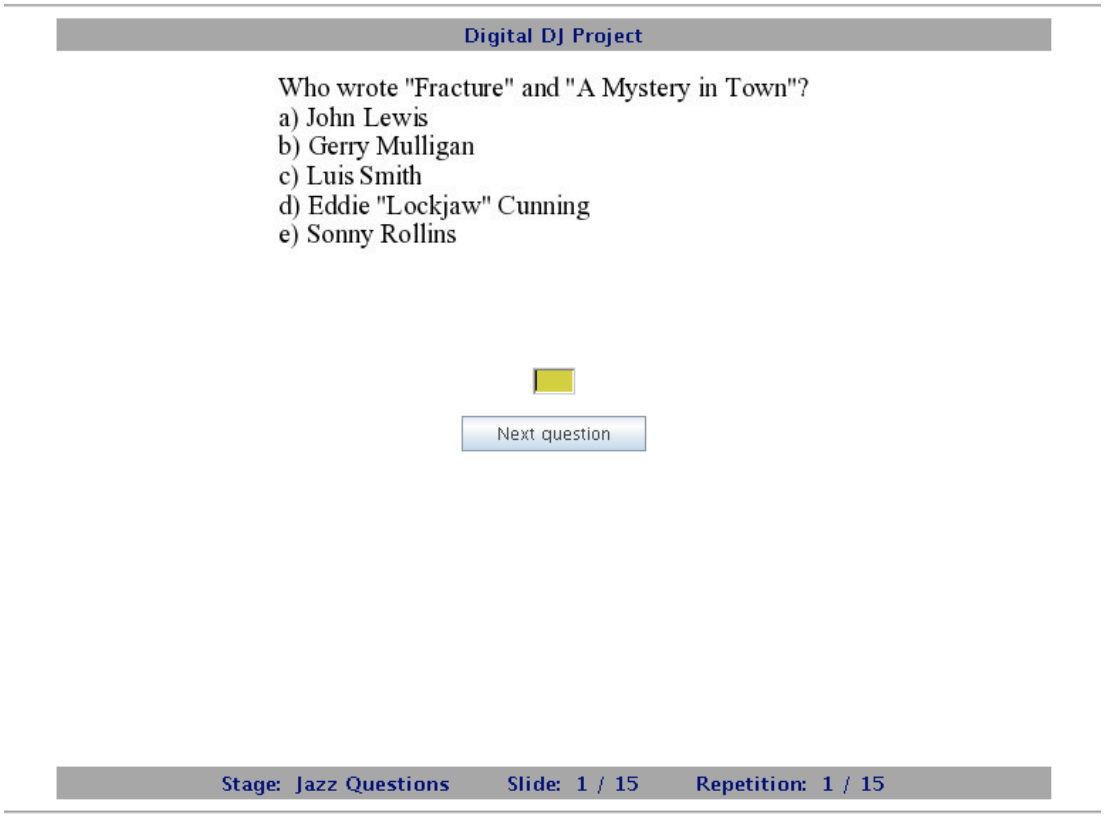

Figure 6.28: Participant's environment within a standard web browser during the portion of our experiment where *factual recall* questions are asked.

Responses to these questions will test our hypothesis that people learn more from text generated using Methodius' parameterized comparison algorithm than text generated without comparisons. The questions were carefully designed so that questions for both song types would be equally challenging. If we find that participant scores were significantly higher following the text containing comparisons that text without comparisons, we can verify our hypothesis that people learn more from text containing comparisons than text without comparisons.

In order to complete this research in a timely manner, we did not develop a seamless music player interface, as is proposed with the "DJ4me" project [5]. For the pilot experiment, the twelve *post-experiment survey* questions were presented to participants in an email.

Our entire experiment and interface was developed using the XML specification language defined for WebExp2. We developed the experiment primarily by expanding an existing experimental setup provided by the WebExp2 developers [28]. We also used a web server provided by the WebExp2 development team to host our experiment. All of the texts in our experiment were pre-generated to prevent computing resource complications during the experiment. As in the previous study, we did not randomly order these six-song sequences for each individual experiment because it was necessary to select a sequence of music pieces that maximized the number of potential comparisons that could be generated by Methodius.

The WebExp2 interface allowed us to improve upon the experimental design of the previous M-PIRO study. Since all twelve texts about music pieces were presented within the WebExp2 interface, a participant could only view a given generated paragraph once. This prevented participants from re-reading previous paragraphs, or accessing them during *factual recall* stages of the experiment. Unlike the study conducted by Karasimos and Isard, we were able to randomize which text type (either jazz or classical) was seen first by participants. Furthermore, all *factual recall* questions were randomly ordered for each participant to minimize any potential ordering effects that could occur with paper versions of the *factual recall* questions, like in the Karasimos and Isard study [8]. No question in our experiment was left unanswered because the WebExp2 interface ensured that no input textbox was left empty [31]. Unfortunately, this forces participants to guess on *factual recall* questions they cannot recall a solution for. In the Karasimos and Isard study, the experimenter discouraged participants from answering a question unless they could recall its answer. We decided to add an additional answer choice (e) to alleviate this problem.

We expected that the pros of the web interface would outweigh its cons. Participants would be able to access our experiment from any computer with Internet access. However, this also means that we have little control over the participant's environment. To alleviate this problem, we force the experiment to open in a new window, which provides some control to the computer screen that is presenting the experiment. Participants of our experiment must also have Internet access and must be able to operate a web browser enabled with Javascript [31].

6.2.4 Subjects

We personally contacted 4 fluent English speakers (1 female, 3 male) for our pilot study by email. All participants ranged in age from 25 to 38. We only performed this study in order to test that the text flowed naturally and had easily understandable instructions. Thus, we did not collect data for analysis. One of our participants was very knowledgeable in the domain of jazz music, while two participants were moderately knowledgeable in the domain of classical music. Our participants did not suffer from past reading difficulties. All participants completed the experiment successfully.

6.2.5 Procedure

We provided all participants in our pilot study with an email containing a hyperlink to our web experiment and a set of *post-experiment questions* to be answered after they completed the web experiment. In our email, we told participants that the experiment should take approximately 20 minutes to complete, the same amount of time it took to complete the experiment in the Karasimos and Isard study [8].

The hyperlink contained in the email pointed to a webpage on the WebExp2 server that provided participants detailed instructions on how to take the experiment. The instructions first detailed the stages of the experiment, ranging from the two text-reading stages, the two *factual recall* stages, and the *post-experimental survey*. The *factual recall* stages were described as "What did you learn?" stages of the experiment. The number of text paragraphs to be read, along with the number of questions to be answered, was provided in the instructions. Two initial stages, a "user input" stage and a "practice" stage were also mentioned in the instructions. Participants were informed that any information they provide about themselves would not be associated with their identity. In addition, they were told that they could exit the experiment at any time without penalty, although they were encouraged to finish. When the participant was ready to begin the experiment, they could click the "START!" hyperlink on the webpage. A copy of the instructions can be found in the Appendix.

Once the participant began the experiment, they must first proceed through two preliminary stages. During the first stage, the participant filled in the fields for name, email address, age, sex, occupation, and native languages. Every field must be completed in order for the participant to proceed to the next stage. The fields for "age" (digits) and "sex" ("m or M" for male and "f or F" for female) were restricted to guarantee consistent input. The participant could press the "SUBMIT" button to proceed to the "practice" stage of the experiment.

The participant proceeded to two webpages that represented the "practice" stage of the experiment. The first page resembled a typical page containing a generated paragraph. The participant could press the "Next" button to proceed. The second page resembled a typical page containing a multiple-choice question. To help the participant gain familiarity with the interface, he or she must have entered any letter "a" through "e" in order to proceed with the actual experiment. Participants were not timed during the actual experiment, however their information was time-stamped to discourage any unreasonably fast progressions through the experiment. All of their input information was coded for confidentiality and stored on the WebExp2 server, except for the *post-experimental survey*, which was emailed back to the experimenter. For the pilot study, we asked participants to leave comments for any potential improvements in their email reply as well.

6.3 Discussion of Feedback

Following the pilot study, we found that a few elements of our experiment could be improved. Our participants all told us the multiple-choice questions in the experiment were difficult to answer, which verified that we made our experiment more challenging than the Karasimos and Isard study [8]. We were also told that the number of facts per paragraph about a music piece was too large (15 facts per paragraph). We addressed this issue by lowering the maximum number of facts per paragraph to 9 in our main experiment.

To maintain confidentiality for all participant input, we decided to add the *post-experimental survey* to the WebExp2 interface for the main experiment. Also, one of our participants pointed out that the factual relationship of one performer being "followed by" another performer was confusing and uninteresting, so we removed the relation from our knowledge base. To balance out the multiple-choice questions for jazz music, we changed the question asking about the "followed-by" relationship to a question asking about the writer of a jazz piece. The pilot study was also able to verify that participant information was being stored on the WebExp2 server successfully.

Chapter 7

Primary Experiment

7.1 Introduction

Our primary experiment strongly resembled our pilot study. We implemented a few modifications based on the suggestions proposed by the participants of the pilot experiment. The principal changes were that we added the *post-experimental survey* to the web experiment, and reduced the maximum number of facts in a generated paragraph to 9 (from 15). Our goal for the primary experiment is to observe a significant difference in performance in the seven "COMPARISON QUESTION" questions asked for both text types, based on whether the text they read about had comparisons or not. These questions specifically asked about facts in the paragraphs that could be reinforced if the encountered text had comparisons. In other words, we wanted to observe if participants performed significantly better on these questions after reading texts with comparisons versus after reading texts that lacked comparisons.

We would also observe any potential ordering effects that people have, such as if they do worse on the first section, then better on the second section because they know the type of questions that would be asked. We would also like to find out if there's a significant difference in the Likert-scale scores for questions asking about people's perceived improvement of learning based on the text type they encountered. In other words, we wanted to see if people subjectively rated texts better when the texts they read had comparisons in them. For example, an ideal participant assigned to one of the two conditions that had comparisons generated for jazz music would like the jazz texts much better and performed much better on the jazz multiple-choice questions because that text had comparisons in it, while the classical music lacked comparisons.

7.2 Method

7.2.1 Designing and Selecting the Song Texts

Our only change to the song texts from the pilot experiment was that we reduced the maximum number of facts per generated paragraph to 9. A complete listing of all song texts, both with and without comparisons, can be found in the Appendix.

7.2.2 Designing the Evaluation Questions

We only changed one *factual recall* question from our pilot study. This jazz-based question asked about identifying the "follower" of a particular performer. We replaced this question with one about identifying the "writer" of a particular piece. This also balanced out the types of questions asked across the jazz and classical text types. Our tally table was updated as follows:

Fact Type Queried	Classical	Jazz
Which album is M from?	2	3
Which period was M written during?	2	3
Where did P originate from?	1	1
What instrument did P play?	0	1
Which music pieces were performed by P?	2	2
Which decade or time period was P active during?	1	1
Who wrote/composed M?	2	2
Which record label released A?	1	1
Who served as an influence to P?	1	1
Who conducted M?	2	0
Which pieces were conducted by P?	1	0

Table 7.2: A tally of the types of *factual recall* questions asked for our primary experiment. **Legend:** M = music piece, P = person, A = album.

7.2.3 Designing the Web Interface with WebExp2

We updated our web interface by adding the twelve *post-experimental survey* questions to the end of the experiment in the same design as the *factual recall* questions. These questions were not presented randomly because they were only used for subjective evaluation. Here we supplied a third webpage type that was displayed after the participant completed the primary stages of the experiment. This set of pages asked the participant questions about the quality of the generated text and how much they had learned from the text. The results of these questions would permit us to evaluate our hypothesis that people judge text generated with Methodius' parameterized comparison algorithm to be more informative than text generated without comparisons [9]. We also asked the user to rate overall the amount learned from each song type (i.e., "I learned a great deal from the jazz texts.") on a Likert scale of 1 (strongly disagree) to 5 (strongly agree) [29]. Results from this rating will evaluate our supplemental hypothesis that people perceive that they learn more from text containing comparisons than from text without comparisons. If we find that participant ratings of song texts were significantly higher for text generated with Methodius' parameterized comparison algorithm than text generated without comparisons, we can verify our additional hypothesis that people found text containing comparisons to be more interesting and enjoyable than text without comparisons.

We also added a free-response question asking for people to select the music type they preferred and to justify their answer. Also, the font format of our texts was streamlined to 14-point Times New Roman. This font is very commonly read on screens and on paper. We selected this font type to best simulate printed paper, as was done for the Karasimos and Isard study [8].

7.2.4 Subjects

Forty participants were recruited for our primary experiment, the same amount of participants in the user study described in [8]. However, in our experiment, all of our participants were fluent English speakers. We decided upon *fluent* English speakers over *native* English speakers because our experiment's text contained relatively simple grammar rules, and primarily contained facts. Since our experiment investigated the recall of facts presented in generated texts, we expected that both fluent and native speakers of English would perform similarly.

In total, fifty-one fluent English speakers took part in our experiment, but several participants' results had to be discarded due to note taking. We were able to acquire this information by emailing all participants once they completed the experiment. We also discarded participants that were experts in jazz or classical music. This information was obtained from a question in the *post-experimental survey*. We assume that participants were honest in their responses. Since there were four experimental conditions to our study, at least 10 participants were required for each condition. This allowed us to compare our results to those of the previous study carried out by Karasimos and Isard [8]. All participants ranged in age from 16 to 62. In total, there were 29 male participants and 11 female participants. Among the 40 fluent English speakers, 27 were native English speakers. Nearly all of the participants had little experience with jazz and classical music. However, some of them have had experience with natural language generation systems. All participants were naïve in reference to our experiment's goals.

We chose to publicize our experiment on the World Wide Web because WebExp2 has several software features that prevent participants from engaging with the system inaccurately. For example, the experiment will timeout if the participant is traversing through the experiment unreasonably quickly (i.e., less than 5 seconds per page). We opened the experiment to public access on several websites [32-34]. The experimenters also sent emails to mailing lists at the University of Edinburgh to announce the experiment. All participants were randomly assigned to 1 of the 4 conditions before they participated in the experiment. All data we collected from these participant evaluations of generated text was stored on the WebExp2 server for data analysis.

7.2.5 Procedure

Since our experiment took place entirely on the web, we could not control the environment of the participants beyond their computer screens. Since many of our participants were Master of Science students in the School of Informatics, one known location that people took part in the experiment was a quiet lab in Appleton Tower.

The instructions for our experiment remained nearly equivalent to those of our pilot study. All participants in our experiment were first directed to a webpage containing the same instructions as our pilot experiment. This webpage included information about the procedure of the experiment. These instructions explicitly told participants that they would be answering questions about "what they learned" from the music texts. Our only modification was that we added a statement about a public prize draw for all participants in the study. Three participants received £25 gift certificates to Amazon.co.uk [35].

The experimental procedure remained largely equivalent to our pilot study. Participants first needed to fill in a webpage requesting their personal details. Next, they went through two practice pages, one symbolizing what a text paragraph would look like and one symbolizing what a multiple-choice question would look like. Since participants were randomly assigned to 1 of the 4 conditions of our experiment, they either read about 6 jazz music pieces or classical music pieces first. Each music piece was described in a single paragraph, with one paragraph per webpage. This first set of texts either had comparisons between music pieces within the texts, or did not have comparisons. After reading the first set of texts, participants answered 15 multiple-choice *factual recall* questions about that text set. They then read about six more music pieces. These pieces would be in the two corresponding text types that were not previously presented. Thus, if the first set of texts presented jazz music pieces with comparisons, the second set of texts presented classical music pieces without comparisons. Participants then answered 15 more multiple-choice *factual recall* questions about the previous text set. Our only change to the overall progression through the experiment was to add the twelve *post-experimental survey* questions and the one free-response question to the end of the web experiment.

7.3 Results/Data Analysis

To extract the data from the WebExp server, we used a sophisticated text editor to enter participants' input into a spreadsheet [36]. Similar to the analyses performed in [8], we tested for significant differences in our data using the two-way repeated measures ANOVA test, which are ideal for experiments with more than 2 conditions. We had access to SPSS 15.0 data analysis software that permitted us to perform this evaluation [37].

In our experiment, we wanted to find significant difference between the "COMPARISON QUESTION" question scores for participants on the texts with comparisons they read, and the "COMPARISON QUESTION" question scores for the texts without comparisons that they read. We divided up our participants according to the between-subjects and within-subjects factors as follows:

Our between-subjects factors:
compGroup - which comparison group they were assigned to (either jazz-with or classical-with)
orderFirst - which music type they heard first (either jazz-then-classical or classical-then-jazz)

Our within-subjects factor:

comparison - 2 levels – Participants' scores on the questions on the text (A) with comparisons and (B) without comparisons.

This meant we combined jazz and classical scores across conditions as appropriate. For example, for the compGroup condition "jazz-with" the jazz-based "COMPARISON QUESTION" values for those 20 participants were merged into the "scores on questions on the texts with comparisons". This also meant that for the compGroup condition "jazz-with" the classical-based "COMPARISON QUESTION" values for those 20 subjects were merged into the "scores on questions on the texts without comparisons".

7.3.1 Preliminary Analyses

Our preliminary analyses showed trends in support of our hypothesis that people will score significantly higher on "COMPARISON QUESTION" questions when the corresponding text has comparisons. In the table below we can see that participants' performance on these questions given the texts had comparisons had an overall mean score of **4.30** (out of 7), a full point higher than the scores of participants on these questions given the texts did not have comparisons (overall mean score of **3.23** overall). To make the participants' performance scores on these questions more viewable, we divided their scores into the following two charts by the music genre that had comparisons.

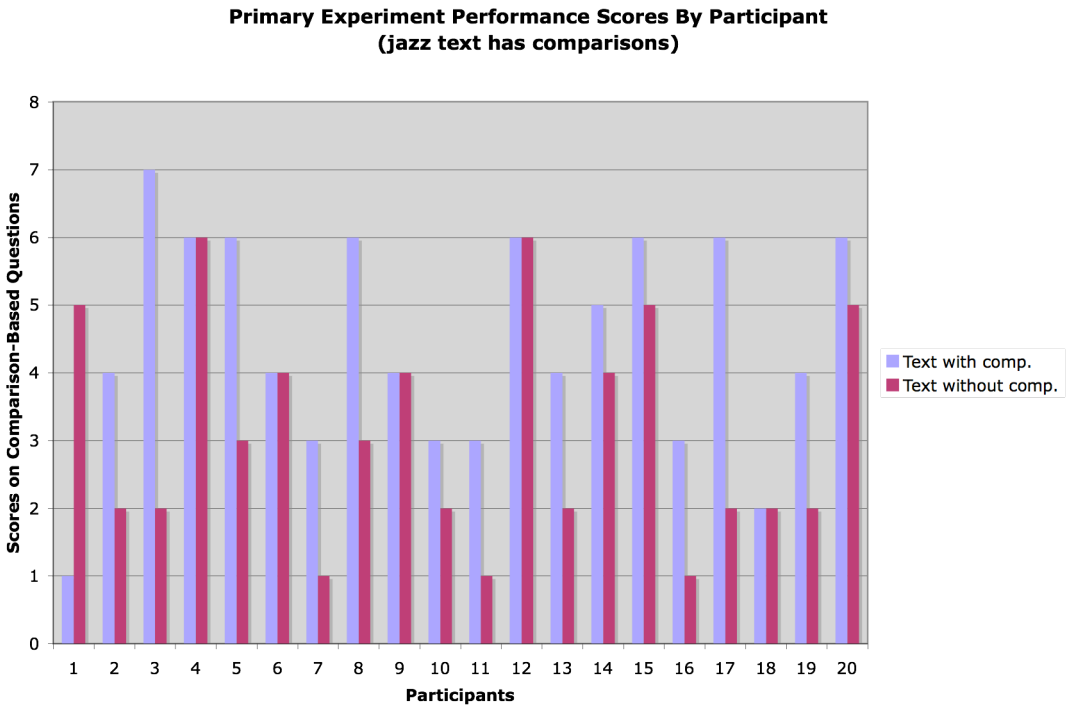

Figure 7.29: Performance scores on “COMPARISON QUESTION” questions by participant depending on the presence of comparisons. Here the jazz text has comparisons.

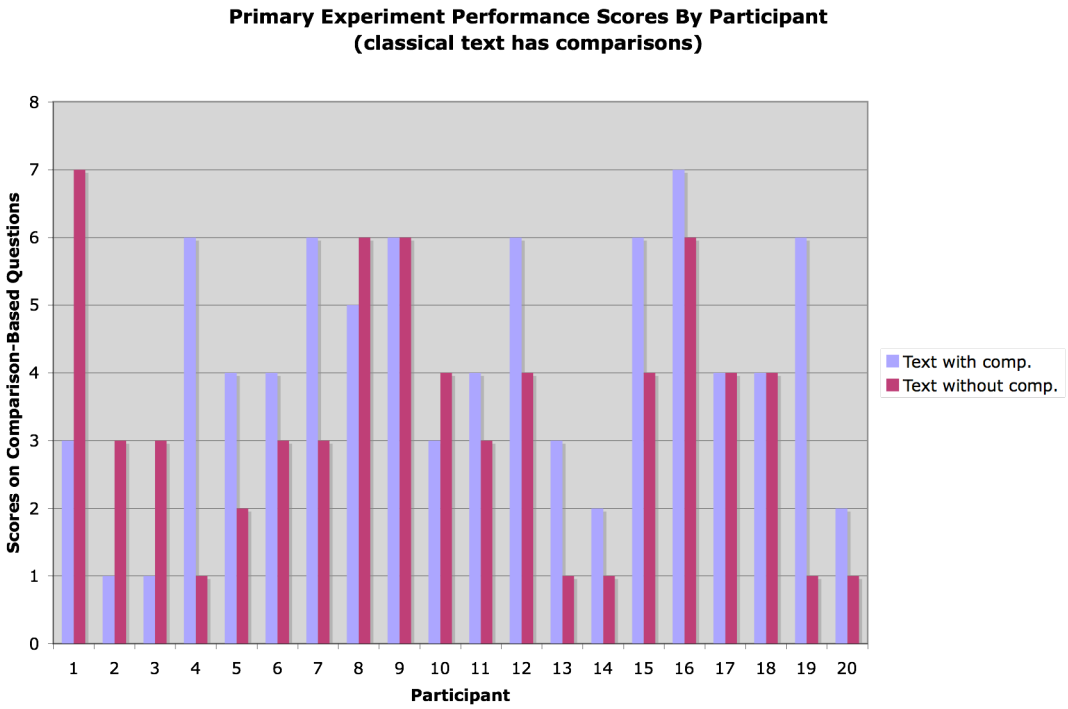

Figure 7.30: Performance scores on “COMPARISON QUESTION” questions by participant depending on the presence of comparisons. Here the classical text has comparisons.

	compGroup	orderFirst	Mean	Std. Deviation	Subject Count
numeric score on CMP questions after reading text **with** comparisons	**Group A** read classical texts with comparisons (jazz without)	classical then jazz	3.9	1.912	10
		jazz then classical	4.4	1.776	10
		Total	**4.15**	1.814	20
	Group B read jazz texts with comparisons (classical without)	classical then jazz	4.5	1.509	10
		jazz then classical	4.4	1.838	10
		Total	**4.45**	1.638	20
	Total	classical then jazz	**4.2**	1.704	20
		jazz then classical	**4.4**	1.759	20
		Overall Mean	**4.3**	1.713	40
numeric score on CMP questions after reading text **without** comparisons	**Group A** read classical texts with comparisons (jazz without)	classical then jazz	3.8	1.932	10
		jazz then classical	2.9	1.792	10
		Total	**3.35**	1.872	20
	Group B read jazz texts with comparisons (classical without)	classical then jazz	3	1.826	10
		jazz then classical	3.2	1.549	10
		Total	**3.1**	1.651	20
	Total	classical then jazz	**3.4**	1.875	20
		jazz then classical	**3.05**	1.638	20
		Overall Mean	**3.23**	1.747	40

Table 7.3: Descriptive statistics for our experiment. CMP questions correspond to the seven questions in each 15-question set whose solutions can be reinforced by text with comparisons.

The table above shows descriptive statistics for combined scores (jazz and classical music pieces) based on participants' *overall* performance on the 7 "COMPARISON QUESTION" questions they were presented after reading 6 generated paragraphs (a) with comparisons and (B) without comparisons. The table also breaks down participants' means by the within-subjects factors of **compGroup** and **orderFirst**.

The compGroup factor indicates whether or not participants were presented with classical texts with comparisons or jazz texts with comparisons. We can see in the above table that the differences of overall means given the compGroup within-subjects factor were relatively similar between Groups A and B (Group A: 4.15 – 3.35 = **0.8**, Group B: 4.45 – 3.1 = **1.35**). Thus, our preliminary analysis suggests that participants will not perform significantly different given that their *jazz* texts had comparisons or their *classical* texts had comparisons.

The orderFirst factor indicates whether or not participants were presented with classical texts first or jazz texts first. We can also observe in the above table that the differences of overall means given the orderFirst within-subjects factor were relatively similar between participants that read classical texts first and those that read jazz texts first (Classical-First: 4.2 – 3.4 = **0.8**, Jazz-First: 4.4 – 3.05 = **1.35**. Thus, our preliminary analysis suggests that participants will not perform significantly different given that they read their *jazz* texts first or their *classical* texts first.

Through our preliminary analyses, we were also able to observe if the questions in one genre were any more challenging than the than those in the other genre. We found that the difficulty of the questions for jazz and classical music pieces was relatively similar. This is because the differences of overall means were similar given the genre of the multiple-choice questions (Classical Questions: 4.15 – 3.1 = **1.05**, Jazz Questions: 4.45 – 3.35 = **1.1**). Our preliminary analysis therefore suggests that participants will not perform significantly different on the *classical* multiple-choice questions versus the *jazz* multiple-choice questions.

7.3.2 Primary Analyses: 2-way repeated measures ANOVAs

Given the setup of our experiment (i.e., 2 between-subjects factors and 1 within-subjects factor) we performed a 2-way repeated measures analysis of variance (ANOVA). We were looking for the repeated measure of "COMPARISON QUESTION" scores *within* participants to overall have a significant difference between participants after reading the texts with comparisons versus participants after reading the texts without comparisons. We declare the following hypotheses that will test the validity of our main hypothesis that people will learn more on texts containing comparisons:

> H_0 (null hypothesis): The performance of the participants in the "COMPARISON QUESTION" section does not depend on the presence of comparisons in the text previously presented.
>
> H_1 (alternative hypothesis): The performance of the participants in the "COMPARISON QUESTION" section **depends** on the presence of comparisons in the text previously presented.

We were able to observe any potential ordering effects, such as whether paying attention more after reading and answering the first round of questions significantly influenced performance on the second part. In addition, we were able to observe any potential grouping effects, such as if participants overall were performing better on one music type (e.g., jazz) over the other (e.g., classical). We also paid attention to mixed grouping and ordering effects. Lastly, the 2-way repeated measures ANOVA permitted us to observe if there were any significant differences in performance on these questions *between* subjects. Ideally, the only significant result should be a statistically significant difference in the "COMPARISON QUESTION" scores between participants after reading texts with comparisons versus participants after reading texts without comparisons. There should be no statistically significant (1) ordering effects or (2) grouping effects. If these two conditions are the case, then we are able to reject the null hypothesis H_0 in favor of the alternative hypothesis H_1.

Our data analyses using the 2-way repeated measures ANOVA confirmed our hypothesis that participants performed significantly better on the "COMPARISON QUESTION" questions given that they previously read text containing comparisons versus participants that previously read text lacking comparisons. The ANOVA revealed that the **comparisons** within-subjects factor was strongly statistically significant ($F = 11.131$, $p < .01$, alpha = .002, df_numerator = 1, df_denominator = 36). Please see the Appendix for a guide for statistical analysis with the 2-way repeated measures ANOVA. The **F-value** tests the null hypothesis H_0. It represents whether the means we are sampling in our ANOVA are *within sampling variability of each other*. A large F-score (i.e., much greater than 1) indicates that we must reject the null hypothesis H_0 [38]. Thus, since our $p < .05$ and $F >> 1$, we were able to reject the null hypothesis H0 in favor of the alternative hypothesis H1 that states that the performance of the participants in the "COMPARISON QUESTION" section **depends** on the presence of comparisons in the text previously presented. Participants performed significantly different on these questions based on whether the texts they read had comparisons. Given our preliminary analysis that the mean score for the "COMPARISON QUESTION" questions was higher after participants read texts containing comparisons over texts lacking comparisons, we can confirm our main hypothesis. People learned more from texts containing comparisons versus text without comparisons.

We found no ordering or grouping effects to be statistically significant. The ANOVA showed that the grouping factor, **comparisons*compGroup**, was not statistically significant because the F-value was less than 1 ($F = .728$, $p > .05$, alpha = .399, df_numerator = 1, df_denomenator = 36). Thus, we cannot reject the null hypothesis. H_0 states that the performance of the participants does not depend on whether the texts with comparisons were jazz or classical.

The ANOVA also showed that the ordering factor, **comparisons*orderFirst**, was not statistically significant because the F-value was less than 1 ($F = .728$, $p > .05$, alpha = .399, df_numerator = 1, df_denomenator = 36). Thus, we cannot reject the null hypothesis. H_0 states that the performance of the participants does not depend on whether the classical or jazz texts were presented first or second. In addition, the ANOVA showed that the mixed grouping-ordering factor, **comparisons*compGroup*orderFirst**, was not statistically significant because the p-value was not less than .05 and the F-value was close to 1 ($F = 1.740$, $p > .05$, alpha = .195, df_numerator = 1, df_denomenator = 36). Thus, we cannot reject the null hypothesis that the **comparisons*compGroup** grouping interaction is the same whether the jazz or classical texts were presented first. Therefore, we can conclude that the performance of the participants does not depend on whether their texts with comparisons were jazz or classical, or on the order in which the texts were presented.

Our ANOVA tests also confirmed that there were no statistically significant between-subjects factors. The **compGroup** between-subjects factor was not significantly different between participants because the F-value was less than 1 ($F = .003$, $p > .05$, df_numerator = 1, df_denomenator = 36). Thus, we cannot reject the null hypothesis, which states that the performance of the participants does not depend on whether the jazz texts or classical texts had comparisons between participants. The **orderFirst** between-subjects factor was not significantly different between participants because the F-value was less than 1 ($F = .027$, $p > .05$, df_numerator = 1, df_denomenator = 36). Thus, we cannot reject the null hypothesis, which states that the performance of the participants does not depend on whether the texts were ordered with jazz first or classical first between participants. The **compGroup*orderFirst** mixed between-subjects factor was not significantly different between participants because the F-value was less than 1 ($F = .074$, $p > .05$, df_numerator = 1, df_denomenator = 36). Thus, we cannot reject the null hypothesis, which states that the performance of the participants does not depend on whether the jazz texts or classical texts had comparisons given the ordering of the texts between participants.

7.3.3 Supplemental Analyses of Post-Experimental Survey Data

In addition to our primary analyses, we tested our supplemental hypothesis that people perceive that they learn more from text that contains comparisons. We focused on the final two questions of the *post-experimental survey*. These questions asked participants to rate the following two statements on a Likert scale of 1 (strongly disagree) to 5 (strongly agree) [29]:

> **Question 1:** "I learned a great deal from the text about jazz music."
>
> **Question 2:** "I learned a great deal from the text about classical music."

Figure 7.31: Post-experimental questions asking about participants' perceptions of learning from texts by their genre.

Our preliminary analyses did not find a significant difference between participants' Likert scale ratings for text with comparisons versus text without comparisons. Overall, when the texts had comparisons, the mean score for perceived learning from music texts was 2.55, which rounds to "neither agree nor disagree" on our Likert scale. As expected, this score was higher, though not significantly higher, than the mean score for perceived learning from music texts that did not contain comparisons, which was 2.48 and rounds to "disagree" on our Likert scale. When the jazz texts had comparisons, the mean score for perceived learning from jazz music texts was 2.20 ("disagree" on our Likert scale). Surprisingly, the mean score for perceived learning from classical music texts without was 2.35, slightly higher than from jazz music texts with comparisons, but on the same point of the Likert scale ("disagree"). When the classical texts had comparisons, the mean score for perceived learning from jazz music texts without comparisons was 2.60 ("neither agree nor disagree" on our Likert scale). As expected, the mean score for perceived learning from classical music texts was slightly higher at 2.90 ("neither agree nor disagree" on our Likert scale). Since in both cases, the mean score for perceived learning from classical music texts was higher than that of the jazz texts, background knowledge may have influenced our results. People in general may have had greater knowledge of jazz music over classical music, which suggests that people would always learn more from the classical music texts.

Since Likert scales are non-parametric, we investigated our hypothesis that people *perceive* that they learn more from texts with comparisons with a series of 2-tailed Spearman correlations [29]. We first found that participants' perception Likert scale scores of learning from texts with comparisons and their scores of learning from texts without comparisons increased or decreased together monotonically. Our hypotheses were as follows:

> H_0: Peoples' perception scores of learning from texts with comparisons and texts without comparisons were not monotonically statistically related.
>
> H_1: Peoples' perception scores of learning from texts with comparisons and texts without comparisons were significantly statistically related in the sense that they decreased together or increased together monotonically.

			felt like learned from text (with comp.)	felt like learned from text (without comp.)
Spearman's rho	felt like learned from text (with comp.)	Correlation Coefficient	1	.770(**)
		Sig. (2-tailed)	.	**.000**
		N	40	40
	felt like learned from text (without comp.)	Correlation Coefficient	.770(**)	1
		Sig. (2-tailed)	**.000**	.
		N	40	40
** Correlation is significant at the 0.01 level (2-tailed).				

Table 7.4: Spearman correlation results for participants' perceived learning scores for (a) texts in their experiment with comparisons and (b) texts in their experiment without comparisons.

In the table above, we found that participants' perceived learning scores (1 to 5) for texts in their experiment with comparisons were very strongly correlated with participants' perceived learning scores for texts without comparisons at the $p < .001$ level. Thus, we reject our null hypothesis H_0 in favor of the alternative hypothesis H_1, which states that participants' perceived learning scores for texts with comparisons and texts without comparisons increased or decreased together monotonically. This suggests that participants tended to enter both perception scores based on their overall learning experience during the experiment, and not on each of the two individual text sets.

We also found that participants' perception Likert scale scores of learning from texts with comparisons did not correlate with their overall scores on the seven corresponding "COMPARISON QUESTION" questions. Here we observed our data for any possible correlation between participants' perception scores and "COMPARISON QUESTION" scores for the text set in their experiment that had comparisons. We defined the following hypotheses:

> H_0: Peoples' perception scores of learning from texts with comparisons and their scores on the "COMPARISON QUESTION" questions were not monotonically statistically related.
>
> H_1: Peoples' perception scores of learning from texts with comparisons and their scores on the "COMPARISON QUESTION" questions were significantly statistically related in the sense that they decreased together or increased together monotonically.

			numeric score on CMP ques with CMP	felt like learned from text (with comp.)
Spearman's rho	numeric score on CMP ques with CMP	Correlation Coefficient	1	-0.097
		Sig. (2-tailed)	.	0.551
		N	40	40
	felt like learned from text (with comp.)	Correlation Coefficient	-0.097	1
		Sig. (2-tailed)	0.551	.
		N	40	40

Table 7.5: Spearman correlation results for participants' numeric scores (out of 7) on "COMPARISON QUESTION" questions and their perceived learning scores for texts with comparisons.

The table above shows that participants' numeric scores on "COMPARISON QUESTION" questions did not correlate with their perceived learning scores (1 to 5) for texts in their experiment with comparisons. Thus, we cannot reject our null hypothesis H_0, which states that these two scores are not monotonically statistically related. This suggests that participants did not perceive that they learned more or less from texts with comparisons given their performance on the "COMPARISON QUESTION" questions from the text set with comparisons.

We also found that participants' perception Likert scale scores of learning from texts without comparisons did not correlate with their overall scores on the corresponding "COMPARISON QUESTION" questions. In this case, we observed our data for any possible correlation between participants' perception scores and "COMPARISON QUESTION" scores for the text set in their experiment that did not have comparisons. We stated the following hypotheses:

> H_0: Peoples' perception scores of learning from texts without comparisons and and their scores on the "COMPARISON QUESTION" questions are not monotonically statistically related.
>
> H_1: Peoples' perception scores of learning from texts without comparisons and their scores on the "COMPARISON QUESTION" questions are significantly statistically related in the sense that they decrease together or increase together monotonically.

			numeric score on CMP ques without CMP	felt like learned from text (without comp.)
Spearman's rho	numeric score on CMP ques without CMP	Correlation Coefficient	1	0.188
		Sig. (2-tailed)	.	0.245
		N	40	40
	felt like learned from text (without comp.)	Correlation Coefficient	0.188	1
		Sig. (2-tailed)	0.245	.
		N	40	40

Table 7.6: Spearman correlation results for participants' numeric scores (out of 7) on "COMPARISON QUESTION" questions and their perceived learning scores for texts without comparisons.

This table indicates that participants' numeric scores on "COMPARISON QUESTION" questions did not correlate with their perceived learning scores (1 to 5) for texts in their experiment without comparisons. We cannot therefore reject our null hypothesis H0 that states that these two scores are not monotonically statistically related. This suggests that participants did not perceive that they learned more or less from texts without comparisons given their performance on the "COMPARISON QUESTION" questions from the text set without comparisons.

Thus, we found no statistically significant correlations that support our hypothesis that people perceive that they learn more from texts with comparisons versus texts without comparisons. We attribute this result to several factors. Since we found questions about jazz texts and questions about classical texts to be approximately equally difficult, the extensive difficulty of all questions may have led participants to simply feel like they did not learn a great deal from either text set. In addition, as previously stated, participants' prior knowledge may influence their responses to questions about their perceived learning from jazz or classical texts. For example, in our experiment, participants may already have a greater awareness with jazz music due to their listening habits.

As an additional set of evaluative information for Methodius, we present the mean scores for the remaining *post-experimental survey* questions in the table below.

Condition	Post-Experimental Question	Jazz Text	Classical Text	Free Response
Jazz texts with comparisons Classical texts without comparisons	I found the text to be interesting.	3 (neither agree nor disagree)	2.73 (neither agree nor disagree)	
	I found the questions to be difficult.	3.92 (agree)	4 (agree)	
	I am an expert in this music genre	1.46 (strongly disagree)	1.42 (strongly disagree)	
	I enjoyed reading about these songs.	3.19 (neither agree nor disagree)	3.26 (neither agree nor disagree)	
	I was able to answer some of the questions in this genre without reading the texts.	1.46 (strongly disagree)	1.5 (disag.)	
	Which text (quality, fluency) did you like more?			Jazz (14 out of 20)
Classical texts with comparisons Jazz texts without comparisons	I found the text to be interesting.	2.76 (neither agree nor disagree)	3.12 (neither agree nor disagree)	
	I found the questions to be difficult.	4.32 (agree)	4.12 (agree)	
	I am an expert in this music genre	1.2 (strongly disagree)	1.44 (strongly disagree)	
	I enjoyed reading about these songs.	2.92 (neither agree nor disagree)	3.28 (neither agree nor disagree)	
	I was able to answer some of the questions in this genre without reading the texts.	1.4 (strongly disagree)	1.6 (disag.)	
	Which text (quality, fluency) did you like more?			Classical (12 out of 20)

Table 7.7: Post-experimental questionnaire results by group.

The table above shows the remaining mean scores from the *post-experimental survey*. Participants were primarily neutral in their feelings toward the texts on average. On average, participants neither agreed nor disagreed with the statement that the texts were interesting. Participants also felt neutral about their enjoyment of the texts. Thus, we cannot prove our additional hypothesis that participants would find texts containing comparisons to be more interesting and enjoyable than texts that did not. As the above means indicate, participants were not experts in either jazz or classical music, and could not answer any of the questions using prior knowledge. We removed participants whose score for either of these two questions was above the Likert scale value of 3 (neither agree nor disagree). As expected, people more often preferred the text that had comparisons in the free-response question.

Chapter 8

Discussion and Conclusion

8.1 Results Interpretation

Our primary experiment investigated whether the presence of comparisons in generated texts improved participants' performance on *factual recall* questions. We hypothesized that participants would perform significantly better on the set of seven "COMPARISON QUESTION" questions given that they previously read text containing comparisons versus participants that previously read text lacking comparisons. Our analysis using 2-way repeated measures ANOVAs confirmed this hypothesis. There was a statistically significant difference in the "COMPARISON QUESTION" scores between participants after reading texts with comparisons and participants after reading texts without comparisons. Furthermore, we found no statistically significant grouping or ordering effects.

This result supports previous literature in human learning and generation that states that comparisons in text improve people's learning ability on factual recall tasks. Rumelhart and Norman conducted an experiment investigating how people could learn about word processors. They stated that devising comparisons between word processors and similar objects such as typewriters and tape recorders were crucial in the learning process [39]. The TEXT system was one of the first generation systems that generated comparisons in its text to improve fact retention [11].

The success of comparisons between entities in text has also been shown to be dependent on what the reader already knows about the entities. Milosavljevic argues that where appropriate, readers should be learning from *grounded comparisons*, comparisons that explain any newly encountered entities [40]. If the reader is already familiar with all entities being compared, *grounded comparisons* are not necessary. She explained that comparisons can help people learn more by relating new entities to those previously mentioned [41]. Comparisons have also generally been shown to be helpful when describing entities and achieving communicative goals [42]. In addition, comparisons have been shown to improve learning by correcting the reader's misinterpretations of facts in generated texts [43].

Since our experiment's results support our main hypothesis that people learned more from texts with comparisons than from texts without comparisons, we have also found Methodius' parameterized comparison generation algorithm to be successful in generating comparisons for our study. Methodius' algorithm resulted in meaningful comparisons that influenced how people learned facts about music pieces. In addition, our results support our claim that Methodius generalizes across domains. Methodius has been shown to operate successfully in the new domain of knowledge about music pieces, artists, and time periods.

We carefully designed our experiment so that it would be as directly comparable to past experiments in comparison generation as possible. An experimental evaluation of the one of Methodius' predecessors, the ILEX (Intelligent Labeling Explorer) system, found that text tailored to a user's browsing history, including comparisons between the current object and previously encountered objects, did not improve participants' factual recall tests versus static text [19]. Although our study and theirs did not produce similar results, we argue that our more modern system featuring aggregation yielded comparisons in text that influenced participants' ability to recall the facts they were presented.

As expected, our study's results paralleled those of the M-PIRO evaluation study conducted by Karasimos and Isard [8]. Both our study and theirs found that people learned more from the text that was the most enhanced. In their study, they found people performed best on factual recall tests after reading texts that contained comparisons and aggregations of facts versus texts that contained neither. However, in their study, the same genre (ancient Greek coins) was always presented before the other genre (ancient Greek vessels). They explained that this was done because of pilot participants complaining of the difficulty of the vessels texts. Thus, participants were only assigned to 1 of 2 conditions, either (a) the coins texts had comparisons and aggregated facts or (b) the vessels texts had comparisons and aggregated facts. This ordering effect may have been a flaw in their experimental design.

In our study, we counterbalanced participants across all 4 possible presentations of the music texts. Hence, participants were assigned to 1 of 4 conditions, (a) the jazz music texts had comparisons and came first, followed by classical music texts without comparisons, (b) the classical music texts had comparisons and came first, followed by jazz music texts without comparisons, (c) the jazz music texts did not have comparisons and came first, followed by classical music texts with comparisons, and (d) the classical music texts did not have comparisons and came first, followed by jazz music texts with comparisons. This yielded a stronger experimental design because we found no grouping or ordering effects to be statistically significant. Our design also guarded against any memory effects. For example, participants may pay much more attention to the second set of music texts after answering the difficult series of questions that followed the first set of music texts. This is because they will have a better idea of the kinds of *factual recall* questions they would be asked.

Most *factual recall* multiple-choice questions in the Karasimos and Isard study provided participants with four choices. We decreased the chance that participants would answer the *factual recall* questions correctly through random guessing by adding a fifth choice to the set of potential answers for each question. Also, every question in our experiment only offered participants five choices, of which one was the correct answer. Two questions in each set of factual recall questions in the experiment conducted by Karasimos and Isard required participants to enter two answers.

8.2 Suggestions for Improvements

Although we achieved our desired results for the experiment's main hypothesis, some improvements could be made. In our experiment's instructions, we made no mention that participants could or could not take notes during the experiment. Unfortunately, because of this, some participants felt the need to take notes about the facts presented in the music texts. The *factual recall* multiple-choice questions became extremely easy for note-takers to answer. At the close of the experiment, we sent an email to each participant confirming their completion of the study and asking if they took notes. As previously stated, we discarded the results of participants who responded saying that they took notes. We decided not to put an indication in the instructions barring participants from taking notes because it could potentially influence their results in the experiment. Also, we did not want to directly give participants the idea that note taking was possible, because participants frustrated with the difficulty of the questions could have begun taking notes. Since our experiment was based entirely on the web, we could not control a participant's environment.

A participant's background knowledge about jazz or classical music may have also influenced the results of our study. We cannot say with confidence that participants had absolutely no knowledge about the performing artists and composers in the music texts. Participants' emotional states during the experiment could be influenced by their preference toward one genre of music over the other. In addition, a participant's familiarity with a certain performer or composer may influence how attentive he or she is while reading a given music text. To safeguard against background knowledge seriously influencing the results of our study, we asked participants if they were experts in either music genre in the *post-experimental survey* section of the experiment. We also asked participants if they could answer any of the *factual recall* questions without having read the music texts. We discarded the results of any participant that answered "yes" to either of these questions. We also decided not to use fabricated proper names for factual information in this study because it would not result in the most genuine possible comparisons.

Many participants in our experiment mentioned in the free-response portion of *post-experimental survey* that the *factual recall* questions were very difficult. This can also be seen in our analysis of the *post-experimental survey* Likert-scale question about the perceived difficulty of the *factual recall* questions. As we have seen in the previous chapter, the mean scale score for the statement "I found the (jazz or classical) questions to be difficult" was "agree". Our goal in the design of the *factual recall* questions was to make them moderately difficult so that we could observe a

noticeable improvement in performance if participants performed better after reading texts that contained comparisons. However, since our experiment required participants to answer every *factual recall* question to complete the experiment, the difficult questions forced participants to guess whenever they did not know the answer to a question. In our experiment, the *factual recall* questions had five potential answers. One potential improvement to this could be to add a "None of the above" answer in the place of one of the five potential answers. This may have guarded more effectively against random guessing. Since all of the *factual recall* questions simply asked for participants to recall a certain fact from one of the music texts, we feel that we could not have made the questions themselves much easier.

One surprising result of our experiment was that we found our supplemental hypothesis to be false. Participants did not perceive that they learned more from texts that contained comparisons versus texts that lacked comparisons. We also found that participants did not find the texts containing comparisons to be more enjoyable or interesting than texts without comparisons, one of our additional hypotheses. We suggest that one potential factor that yielded these results was the wording of the *post-experimental survey* questions. Perhaps asking participants to answer how strongly they agreed with the statements "I learned a great deal from the text about (jazz or classical) music" or "I enjoyed reading about the (jazz or classical) music" was too strong considering most participants found our *factual recall* questions difficult. One possible improvement would have been to instead ask participants how strongly they agreed with the statement "I found the text about (jazz or classical) music easy to remember". Instead, we could have asked participants to directly compare the two music texts by asking "Which set of texts did you find easier to remember?" or "Which set of texts did you find you learned more from?". We could have asked participants to respond to one of three choices: (a) jazz, (b) classical, or (c) neither. This may have yielded results that were greater in support for our supplemental and additional hypotheses.

8.3 Future Work

The results of our experiment confirmed that Methodius could be applied to the music domain. We suggest that future work should involve integrating Methodius' generation engine into a sophisticated, customizable digital disc jockey application, as is proposed in the Edinburgh-Stanford Link's "DJ4me" project [5]. This work first would involve the development of a graphical user interface to such a system, followed by a usability evaluation of it. Once this is complete, we anticipate that the "DJ4me" application would then be integrated with the allmusic.com database, which includes facts on hundreds of thousands of musicians and albums, including pictures of artists and their albums [44]. This may require the development of a new knowledge base authoring tool that exports data in a format suitable as input to Methodius. We would like to see how the presentation of visual images and music would influence the results of our experiment. People's like or dislike of the current song being played may influence how much they will remember about the music information that is being presented. In addition, people may or may not find the addition of a picture of the current song's album cover helpful in remembering the facts they are reading about. Integration with the allmusic.com database would extend the number of possible music genres beyond only jazz and classical music.

The "DJ4me" interface could also be enhanced with dialogue interaction. For instance, the user could enter his or her preferences for the type of facts they find interesting, along with preferred music genres, artists, or albums. Future research should also investigate whether the integration of a speech synthesizer as the disc jockey would improve the interaction. Instead of the user reading about generated information about music, the user would listen to a customized radio station and disc jockey. This would require an evaluation study to investigate the influence of a synthesized voice on people's perceived enjoyment with the "DJ4me" application.

In our study, we found that adults that were not experts in jazz or classical music learned more from texts that contained comparisons. In future work, we would like to add user models for other audiences, such as children and music experts. These models would adjust the number of facts mentioned per sentence. In addition, we expect that Methodius will also be extended to tailor the level of fact detail that the system would make based on the expertise of the audience. This type of tailoring is currently being implemented in spoken dialogue systems [45-48].

8.4 Conclusion

This thesis performed an evaluation of the Methodius Natural Language Generation System's parameterized comparison generation algorithm. For our study, we generated texts about music because Methodius may soon be integrated into a customizable disc jockey application. In order to gain a sense of what disc jockeys discussed about music pieces, we transcribed a number of disc jockeys from a classical and jazz radio station. We then authored a knowledge base of facts about music pieces based on the types of facts disc jockeys frequently discussed. This required the development of an ontology of the music domain. We also populated the knowledge base with factual information about music pieces acquired from the allmusic.com database [44].

We conducted an experiment to test several hypotheses that evaluated comparison generation in Methodius. To accomplish this, we developed and executed a web experiment where participants read a number of paragraphs about jazz and classical music pieces. The primary purpose of our experiment was to test whether people learned more from texts containing comparisons produced by Methodius versus texts that did not contain comparisons. After reading a series of six paragraphs about music from one genre, participants answered a series of *factual recall* questions that investigated how well people remembered the facts they were told about. Depending on the participant's assigned condition, this first series of texts either did or did not contain comparisons. Participants then read a series of six more paragraphs about music pieces that contained comparisons if their first series of texts lacked comparisons, or vice versa. They then answered another set of *factual recall* questions. Afterwards, participants answered a *post-experimental survey* assessing their subjective opinions of the generated texts.

Our results confirmed our main hypothesis, which stated that people would learn more from texts that contained comparisons versus texts that lacked comparisons. These results also verified that Methodius' parameterized comparison generation algorithm could generalize to the music domain. However, participants' subjective responses did not confirm our supplemental hypothesis. We could not confirm that people perceived that the learned more from texts that contained comparisons versus texts that did not. Also, people found the texts containing comparisons as interesting and enjoyable as the texts lacking comparisons. We propose that future studies further investigate people's subjective opinions of Methodius' generated texts. We hope that this evaluation of comparison generation will lead to improvements in future natural language generation systems.

Appendix A

Texts with Comparisons

Classical Music Texts

"Molto Moderato" was from the album "Sonate B"; it was performed by Valery Afanassiev and it was composed by Franz Schubert, who was active during the early 19th century. Franz Schubert originated from Vienna, Austria and he was influenced by Ludwig van Beethoven. "Molto Moderato" was conducted by Paul Westwood, who originated from Philadelphia, Pennsylvania, and it was written during the Romantic period.

Like "Molto Moderato", "The Great" was written during the Romantic period and it was composed by Franz Schubert. It was performed by the Royal Cambridgeshire Orchestra and it was conducted by Nikolaus Harnoncourt, who was active during the late 20th century. Nikolaus Harnoncourt originated from Berlin, Germany. "The Great" was from the album "The Symphonies", which was released on the Teldec label. The album "The Symphonies" was originally recorded in 1993.

Unlike the symphonies you recently read about, which were written during the Romantic period and were composed by Franz Schubert, "Liberte" was written during the Post-modern Classical period and it was composed by Francis Poulenc. Poulenc originated from Paris, France; he was active during the mid-20th century and he was influenced by Eric Satie. "Liberte" was conducted by Nikolaus Harnoncourt; it was performed by the Royal Cambridgeshire Orchestra and it was from the album "Figure Humaine", which was originally recorded in 1990.

Unlike "Liberte" and "The Great", which were conducted by Nikolaus Harnoncourt, "Daphnis et Chloe" was conducted by Bruno Walter and it was composed by Maurice Ravel. Bruno Walter originated from Berlin, Germany. Maurice Ravel originated from Paris, France; he was active during the early 20th century and he was influenced by Eric Satie. "Daphnis et Chloe" was written during the Modern Classical period; it was from the album "La Mer" and it was performed by the Boston Symphony Orchestra.

Like the previous orchestral piece, "Prelude to the Afternoon of a Faun" was from the album "La Mer", which was originally recorded in 1999. It was conducted by Charles Munch and it was composed by Claude Debussy, who was influenced by Eric Satie. Claude Debussy originated from St. Germain-en-Laye, France and he was active during the late 19th century. "Prelude to the Afternoon of a Faun" was written during the Impressionist period and it was performed by the Orchestre National de France.

Unlike the previous two orchestral pieces you recently read about, which were from the album "La Mer", "Adagietto" was from the album "Symphony No. 5", which was originally recorded in 1938. It was conducted by Charles Munch and it was composed by Gustav Mahler, who was active during the late 19th century. Gustav Mahler originated from Vienna, Austria and he was influenced by Richard Wagner. "Adagietto" was written during the Romantic period and it was performed by the Vienna Philharmonic.

Jazz Music Texts

"Fracture" was from the album "Nostalgia" and it was performed by Fats Navarro, who was active during the 1940s. Fats Navarro originated from Key West, Florida; he played the trumpet and he was influenced by Roy Eldridge. "Fracture" was written during the Bebop period by Eddie "Lockjaw" Cunning, who originated from New York City, New York.

Like "Fracture", "Jeru" was written during the Bebop period. It was from the album "Legacy"; it was written by Gerry Mulligan and it was performed by Miles Davis, who was influenced by Roy Eldridge. Miles Davis played the trumpet and he originated from Alton, Illinois; he participated in the Miles Davis Quintet and he was active from the 1940s to the 1980s.

Unlike the bop style pieces you recently read about, which were written during the Bebop period, "A Mystery in Town" was written during the Cool Jazz period. It was written by Eddie "Lockjaw" Cunning, who was active from the 1940s to the 1990s, and it was performed by Fats Navarro. Navarro was influenced by Roy Eldridge, who originated from Pittsburgh, Pennsylvania. "A Mystery in Town" was from the album "Double Talk", which was released on the History label. The album "Double Talk" was originally recorded in 1949.

Unlike "Fracture" and "A Mystery in Town", which were written by Eddie "Lockjaw" Cunning and were performed by Fats Navarro, "Avatar" was written by Gary Husband and it was performed by Billy Cobham. Cobham originated from Panama City, Panama and he played the drums; he was active from the 1970s to the 1990s and he participated in the Mahavishnu Orchestra. He was influenced by Miles Davis. "Avatar" was written during the Fusion period and it was from the album "The Promise".

Like "Avatar", "Django" was written during the Fusion period and it was from the album "The Promise". It was written by John Lewis and it was performed by John McLaughlin, who was influenced by Miles Davis. John McLaughlin played the guitar and he originated from Yorkshire, England; he was active from the 1960s to the 1990s and he participated in the Mahavishnu Orchestra.

Unlike the fusion pieces you recently read about, which were written during the Fusion period and were from the album "The Promise", "Alarm" was written during the Free Jazz period and it was from the album "Short Tales". The album "Short Tales" was originally recorded in 1968. "Alarm" was written by John Lewis and it was performed by Frank Lowe, who was influenced by John Coltrane. Frank Lowe played the saxophone; he originated from Memphis, Tennessee and he was active from the 1960s to the 1990s.

Appendix B

Texts without Comparisons

Classical Music Texts

"Molto Moderato" was from the album "Sonate B"; it was performed by Valery Afanassiev and it was composed by Franz Schubert, who was active during the early 19th century. Franz Schubert originated from Vienna, Austria and he was influenced by Ludwig van Beethoven. "Molto Moderato" was conducted by Paul Westwood, who originated from Philadelphia, Pennsylvania, and it was written during the Romantic period.

"The Great" was composed by Franz Schubert and it was performed by the Royal Cambridgeshire Orchestra. It was written during the Romantic period and it was conducted by Nikolaus Harnoncourt, who was active during the late 20th century. Nikolaus Harnoncourt originated from Berlin, Germany. "The Great" was from the album "The Symphonies", which was released on the Teldec label. The album "The Symphonies" was originally recorded in 1993.

"Liberte" was conducted by Nikolaus Harnoncourt and it was composed by Francis Poulenc, who was influenced by Eric Satie. Francis Poulenc originated from Paris, France and he was active during the mid-20th century. "Liberte" was written during the Post-modern Classical period; it was performed by the Royal Cambridgeshire Orchestra and it was from the album "Figure Humaine", which was originally recorded in 1990.

"Daphnis et Chloe" was written during the Modern Classical period; it was performed by the Boston Symphony Orchestra and it was composed by Maurice Ravel, who was active during the early 20th century. Maurice Ravel originated from Paris, France and he was influenced by Eric Satie. "Daphnis et Chloe" was conducted by Bruno Walter, who originated from Berlin, Germany, and it was from the album "La Mer".

"Prelude to the Afternoon of a Faun" was written during the Impressionist period; it was performed by the Orchestre National de France and it was composed by Claude Debussy, who was influenced by Eric Satie. Claude Debussy originated from St. Germain-en-Laye, France and he was active during the late 19th century. "Prelude to the Afternoon of a Faun" was conducted by Charles Munch, who originated from Strasbourg, France, and it was from the album "La Mer".

"Adagietto" was from the album "Symphony No. 5"; it was performed by the Vienna Philharmonic and it was composed by Gustav Mahler, who was active during the late 19th century. Gustav Mahler originated from Vienna, Austria and he was influenced by Richard Wagner. "Adagietto" was conducted by Charles Munch, who was active during the mid-20th century, and it was written during the Romantic period.

Jazz Music Texts

"Fracture" was from the album "Nostalgia" and it was performed by Fats Navarro, who was active during the 1940s. Fats Navarro originated from Key West, Florida; he played the trumpet and he was influenced by Roy Eldridge. "Fracture" was written during the Bebop period by Eddie "Lockjaw" Cunning, who originated from New York City, New York.

"Jeru" was from the album "Legacy" and it was performed by Miles Davis, who was influenced by Roy Eldridge. Miles Davis played the trumpet and he originated from Alton, Illinois; he participated in the Miles Davis Quintet and he was active from the 1940s to the 1980s. "Jeru" was written during the Bebop period by Gerry Mulligan.

"A Mystery in Town" was written during the Cool Jazz period by Eddie "Lockjaw" Cunning, who was active from the 1940s to the 1990s, and it was performed by Fats Navarro. Navarro was influenced by Roy Eldridge, who originated from Pittsburgh, Pennsylvania. "A Mystery in Town" was from the album "Double Talk", which was released on the History label. The album "Double Talk" was originally recorded in 1949.

"Avatar" was from the album "The Promise" and it was performed by Billy Cobham, who was influenced by Miles Davis. Billy Cobham originated from Panama City, Panama and he played the drums; he was active from the 1970s to the 1990s and he participated in the Mahavishnu Orchestra. "Avatar" was written during the Fusion period by Gary Husband.

"Django" was from the album "The Promise" and it was performed by John McLaughlin, who was influenced by Miles Davis. John McLaughlin played the guitar and he originated from Yorkshire, England; he was active from the 1960s to the 1990s and he participated in the Mahavishnu Orchestra. "Django" was written during the Fusion period by John Lewis.

"Alarm" was written during the Free Jazz period by John Lewis and it was performed by Frank Lowe, who was influenced by John Coltrane. Frank Lowe played the saxophone; he originated from Memphis, Tennessee and he was active from the 1960s to the 1990s. "Alarm" was from the album "Short Tales", which was originally recorded in 1968.

Appendix C

Factual Recall Questions

Questions about Classical Music Texts

Who conducted "Molto Moderato"?
a) Charles Munch
b) Nikolaus Harnoncourt
c) Georg Solti
d) Bruno Walter
e) Paul Westwood

Which piece was on the album "Sonate B"?
a) "Molto Moderato"
b) "Adagietto"
c) "Prelude to the Afternoon of a Faun"
d) "Daphnis et Chloe"
e) "Liberte"

Which pieces were composed by Franz Schubert?
a) "Prelude to the Afternoon of a Faun" and "The Great"
b) "Adagietto" and "Daphnis et Chloe"
c) "Liberte" and "Molto Moderato"
d) "Daphnis et Chloe" and "Prelude to the Afternoon of a Faun"
e) "Molto Moderato" and "The Great"

Which period were "Molto Moderato" and "The Great" from?
a) the Postmodern Classical period
b) the Classic period
c) the Baroque period
d) the Romantic period
e) the Impressionist period

Which album was the piece "The Great" from?
a) "The Symphonies"
b) "Sonate B"
c) "Symphony No. 5"
d) "Figure Humaine"
e) "La Mer"

Which period was the piece "Liberte" written during?
a) the Baroque period
b) the Post-modern Classical period
c) the Impressionist period
d) the Romantic period
e) the Medieval period

Who was an influence for both Francis Poulenc and Maurice Ravel?
a) Richard Wagner
b) Gustav Mahler
c) Eric Satie
d) Edward Elgar
e) Franz Joseph Haydn

Which pieces were conducted by Nikolaus Harnoncourt?
a) "Daphnis et Chloe" and "Adagietto"
b) "Liberte" and "The Great"
c) "Adagietto" and "Daphnis et Chloe"
d) "Molto Moderato" and "Liberte"
e) "Prelude to the Afternoon of a Faun" and "The Great"

Where did conductor Bruno Walter originate from?
a) Berlin, Germany
b) Paris, France
c) Philadelphia, Pennsylvania
d) Strasbourg, France
e) Vienna, Austria

Who performed the piece "Daphnis et Chloe"?
a) the Royal Cambridgeshire Orchestra
b) the Orchestre National de France
c) Valery Afanassiev
d) the Boston Symphony Orchestra
e) the Vienna Philharmonic

Which album were "Daphnis et Chloe" and "Prelude to the Afternoon of a Faun" from?
a) "Figure Humaine"
b) "The Symphonies"
c) "Symphony No. 5"
d) "Sonate B"
e) "La Mer"

Who composed "Prelude to the Afternoon of a Faun"?
a) Claude Debussy
b) Franz Schubert
c) Gustav Mahler
d) Francis Poulenc
e) Maurice Ravel

Which album was "Adagietto" from?
a) "La Mer"
b) "Symphony No. 5"
c) "Sonate B"
d) "The Symphonies"
e) "Figure Humaine"

Who performed the piece "Adagietto"?
a) the Boston Symphony Orchestra
b) the Royal Cambridgeshire Orchestra
c) Valery Afanassiev
d) the Orchestre National de France
e) the Vienna Philharmonic

Who conducted "Daphnis et Chloe"?
a) Charles Munch
b) Paul Westwood
c) Bruno Walter
d) Georg Solti
e) Nikolaus Harnoncourt

Questions about Jazz Music Texts

Which album was the song "Fracture" on?
a) "Legacy"
b) "The Promise"
c) "Nostalgia"
d) "Double Talk"
e) "Just Friends"

Which period were "Fracture" and "Jeru" from?
a) the Bebop period
b) the Dixieland period
c) the Free Jazz period
d) the Fusion period
e) the Cool Jazz period

Where did Miles Davis originate from?
a) Philadelphia, Pennsylvania
b) Detroit, Michigan
c) Memphis, Tennessee
d) Alton, Illinois
e) Kent, England

Which period was "A Mystery in Town" written during?
a) the Bebop period
b) the Cool Jazz period
c) the Fusion period
d) the Free Jazz period
e) the Dixieland period

Which instrument did Fats Navarro play?
a) the saxophone
b) the drums
c) the trumpet
d) the piano
e) the guitar

Which songs were performed by Fats Navarro?
a) "Alarm" and "Django"
b) "Fracture" and "Django"
c) "Avatar" and "Alarm"
d) "Fracture" and "A Mystery in Town"
e) "Avatar" and "A Mystery in Town"

Which instrument did Billy Cobham play?
a) the drums
b) the guitar
c) the saxophone
d) the trumpet
e) the piano

Which period were "Django" and "Avatar" written during?
a) the Free Jazz period
b) the Bebop period
c) the Ragtime period
d) the Fusion period
e) the Cool Jazz period

Which decades were John McLaughlin active during?
a) 1910s to 1920s
b) 1920s to 1940s
c) 1940s to 1950s
d) 1950s to 1960s
e) 1960s to 1990s

Who wrote the song "Avatar"?
a) John Lewis
b) Eddie "Lockjaw" Cunning
c) Gary Husband
d) Gerry Mulligan
e) Earl Hines

Which songs were from the album "The Promise"?
a) "A Mystery in Town" and "Django"
b) "Avatar" and "Jeru"
c) "Jeru" and "Alarm"
d) "A Mystery in Town" and "Alarm"
e) "Django" and "Avatar"

Who wrote "Fracture" and "A Mystery in Town"?
a) John Lewis
b) Gerry Mulligan
c) Luis Smith
d) Eddie "Lockjaw" Cunning
e) Sonny Rollins

What album was "Alarm" from?
a) "Nostalgia"
b) "The Promise"
c) "Double Talk"
d) "Legacy"
e) "Short Tales"

Who performed the song "Alarm"?
a) Miles Davis
b) Fats Navarro
c) Billy Cobham
d) Frank Lowe
e) Clifford Thornton

Which artist served as an influence to Frank Lowe?
a) John Coltrane
b) Miles Davis
c) Domenic Trojano
d) Rayford Griffin
e) Roy Eldridge

Appendix D

Post-Experimental Survey Questions

I found the jazz text to be interesting.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I found the classical music text to be interesting.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I found the jazz questions to be difficult.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I found the classical music questions to be difficult.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I am an expert in jazz music.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I am an expert in classical music.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I enjoyed reading about the jazz songs.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I enjoyed reading about the classical music pieces.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I was able to answer some of the jazz questions without reading the texts.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I was able to answer some of the classical questions without reading the texts.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I learned a great deal from the text about jazz music.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

I learned a great deal from the text about classical music.
1) strongly disagree
2) disagree
3) neither agree nor disagree
4) agree
5) strongly agree

Which text (quality, fluency) did you like more and why?

Appendix E

Experiment Instructions

Experiment on Sentences about Music

Experimenter:
Matthew Marge
School of Informatics
University of Edinburgh

Thanks for taking part in this experiment!
Please read the instructions carefully before starting. Do not hesitate to contact the experimenter in case you have any questions or comments concerning this experiment.

PRIZE DRAW: three lucky participants will receive £25 (or $50) gift certificates to Amazon.com! You must complete the experiment in its entirety to be eligible. You will know the experiment is completed when you see the word DONE! across the screen.

Note: Experts in jazz or classical music are discouraged from participating in this experiment.

Technical Requirements

In order to run the experiment, you need Java 1.4 or 1.5.

Instructions

Part 1: Reading sentences about music

We are looking to prototype a Digital DJ system that is intended to discuss a variety of facts about music pieces. In this part of the experiment, you will be presented with 6 paragraphs about pieces of music. Please be sure to read every sentence; you will be asked questions about the music discussed in a later section. For each entry, once you have finished reading the paragraph, please press the NEXT SENTENCE button to procede.

Part 2: What did you learn?

In Part 1 of the experiment you read a sequence of six paragraphs about pieces of music. In this section, you will be presented with 15 multiple-choice questions about the music you just read about. For each question, enter your letter answer in the text box and press the NEXT QUESTION button to procede. Questions require one letter for the answer.

Part 3: More sentences about music
You will read six more paragraphs about pieces of music.

Part 4: What did you learn?
You will answer 15 more multiple-choice questions about the music you just read about in Part 3.

Part 5: Post-Experiment Survey and Feedback Form
You will answer 12 feedback questions about the experiment and are asked to leave some comments.

Procedure

The experiment will consist of the following 4 parts:

Practice session: to become familiar with system
Music Type 1: reading 6 paragraphs, then answering 15 multiple-choice questions about them.
Music Type 2: reading 6 paragraphs, then answering 15 multiple-choice questions about them.
Post-Experiment Survey: answering 12 multipe-choice questions, then filling out a brief comment box form.
The experiment will take about 20 minutes. After the experiment is completed you will receive an email confirmation of your participation.

Your personal details

Before the actual experiment begins, you'll see a form asking for details about yourself (this is the first thing you will see once you've pressed the Start link below). We'd be grateful if you'd give a valid email address so that we can contact you if we have any questions about your answers, and so that we can mail you with information about the purpose of the experiment once it is completed (and if you receive a prize!).

Please be careful to fill in the Personal Details questionnaire correctly, as otherwise we will have to discard your responses. Your information will be kept in the strictest of confidence; all personal information will be coded and your responses in this experiment will not be associated with your personal information. We ask you to supply the following information:

your name and email address;
your age and sex;
your occupation (or student status);
your native languages (e.g., English, Spanish, German, etc.)

Again, the personal data you give us is used only for scientific purposes. We will not give any of this information to anyone else, and nor will we report any information in any way that can be identified with you.

And finally...

Taking part in this experiment is entirely voluntary! Obviously we'd be grateful if you stayed the course, but of course you are at liberty to break off at any point during the experiment.
Once again, thanks for your interest in taking part, and have fun! You can start the experiment proper by pressing on the Start button below.

START!
Experimental design by Matthew Marge and Johanna Moore using WebExp2 Experimental Software

Note: These instructions were largely based off of those composed by Frank Keller for WebExp2 experiments [49].

Appendix F

Statistical Guide

In our experiment, we devised hypotheses that could be rephrased as yes/no (Y/N) questions. For instance, our main hypothesis can be rephrased as "Do people learn more from texts that contain comparisons versus texts that lack comparisons?". We can devise two **competing** hypotheses from this [50]. The **null hypothesis (H_0)** serves as the "control" hypothesis, indicating that a statistically significant result does not occur in either direction of the Y/N answer. The **alternative hypothesis (H_1)** states that a statistically significant result can occur in one or both directions of the Y/N answer, depending on the type of study being conducted. The **independent variable** serves as the possible cause of a result. For instance, in our experiment, the presence and absence of comparisons was an independent variable. The **dependent variable** serves as the possible effect, which is examined by experimenters to produce results to evaluate two competing hypotheses. The "COMPARISON QUESTION" score was one dependent variable in our experiment. Our goal in an experiment is to find a **statistically significant** result that will force us to reject the null hypothesis in favor of the alternative hypothesis. In the statistical tests we conducted, the **p-value** is the probability of producing a result that is *at least* as extreme as a provided data point, assuming that data point did not occur merely by chance [51]. It determines whether we can or cannot reject the null hypothesis. We find a statistically significant result when the p-value is less than .05. A value of $p < .01$ is very significant. When either of these are the case, we can reject the null hypothesis in favor of the alternative hypothesis.

The 2-way repeated measures ANOVA is appropriate for our experiment because we are investigating a repeated measure that is a within-subjects factor. It is a 2-way ANOVA because our experiment has one independent variable, the presence or lack of comparisons. ANOVAs are used when an experiment has more than 2 conditions. We assume that our data has dependent variables that are either of type interval or ratio. We also assume that our data forms a normal distribution and passes the test for homogeneity of variance in order to perform an ANOVA. The **F-value** tests the null hypothesis H_0. It represents whether the means we are sampling in our ANOVA are *within sampling variability of each other*. A large F-score (i.e., much greater than 1) indicates that we must reject the null hypothesis H_0 [38].

Bibliography

[1] G. Carenini and J. D. Moore, "Generating Explanations in Context," in *ACM/SIGCHI International Workshop on Intelligent User Interfaces* Orlando: ACM Press, 1993.

[2] R. Dale, S. J. Green, M. Milosavljevic, C. Paris, C. Verspoor, and S. Williams, "The Realities of Generating Natural Language from Databases," in *11th Australian Joint Conference on Artificial Intelligence* Brisbane, 1998.

[3] M. O'Donnell, C. Mellish, J. Oberlander, and A. Knott, "ILEX: An architecture for a dynamic hypertext generation system," *Natural Language Engineering,* vol. 7, pp. 225-250, 2001.

[4] M. Milosavljevic, "Content Selection in Comparison Generation," in *6th European Workshop on Natural Language Generation (EWNLG)*, 1997.

[5] "DJ4me," The Edinburgh Stanford Link, http://www.edinburghstanfordlink.org/dj4me, 2007.

[6] M. Marge, "Informatics Research Proposal," University of Edinburgh, 2007.

[7] Last.fm, "Last.fm," Last.fm, Ltd. http://www.last.fm.

[8] A. Karasimos and A. Isard, "Multi-lingual Evaluation of a Natural Language Generation System," in *Fourth International Conference on Language Resources and Evaluation (LREC 2004)* Lisbon, Portugal, 2004.

[9] A. Isard, "Choosing the Best Comparison Under the Circumstances," in *International Workshop on Personalization Enhanced Access to Cultural Heritage*, 2007.

[10] A. Melengoglou, I. Androutsopoulos, J. Calder, C. Callaway, R. Clark, A. Dimitromanolaki, I. Hughson, A. Isard, C. Matheson, E. Not, J. Oberlander, D. Spiliotopoulos, S. Varges, and G. Xydas, "Generation Components and Documentation for Prototype D4.5," *M-PIRO Project (IST-1999-10982) Public Deliverable 1.5,* 2002.

[11] K. R. McKeown, "Generating Natural Language Text in Response to Questions about Database Structure," in *Department of Computer and Information Science*. vol. Ph. D. Philadelphia: University of Pennsylvania, 1982.

[12] E. Reiter and R. Dale, "Building applied natural language generation systems," *Natural Language Engineering,* vol. 3, pp. 57-87, 1997.

[13] M. E. Foster, "Evaluating the Impact of Variation in Automatically Generated Multimodal Object Descriptions," in *School of Informatics*. vol. Ph.D. Edinburgh: University of Edinburgh (to appear), 2007.

[14] J. D. Moore, "Content Determination and Structuring: The TEXT System Lecture Slides, Dialogue and Natural Language Generation," Edinburgh: University of Edinburgh, 2007.

[15] C. W. Mann and S. A. Thompson, "Rhetorical structure theory: Towards a functional theory of text organization," *TEXT,* vol. 8, pp. 243-281, 1988.

[16] A. Isard, J. Oberlander, C. Matheson, and I. Androutsopoulos, "Speaking the users' languages," *IEEE Intelligent Systems,* vol. 18, pp. 40-45, 2003.

[17] M. Milosavljevic, "Maximising the Coherence of Descriptions via Comparison." vol. Ph. D. Sydney: Macquarie University, 1999.

[18] C. Mellish and R. Dale, "Evaluation in the Context of Natural Language Generation," *Computer Speech and Language,* vol. 12, pp. 349-373, 1998.

[19] R. Cox, M. O'Donnell, and J. Oberlander, "Dynamic versus static hypermedia in museum education: an evaluation of ILEX," in *Artificial Intelligence in Education Conference*, Le Mans, 1999.

[20] M. White, "Efficient Realization of Coordinate Structures in Combinatory Categorial Grammar," *Research on Language and Computation,* vol. 4, pp. 39-75, 2006.

[21] "BBC Radio Three." vol. 2007 London, England: British Broadcasting Corporation. www.bbc.co.uk/radio3, 2007.

[22] "Rapid Transcription Guidelines." vol. 2007: Linguistic Data Consortium. http://projects.ldc.upenn.edu/Transcription/quick-trans/index.html, 2007.

[23] "Apple Quicktime 7," Apple Inc. http://www.apple.com/quicktime/, 2007.

[24] D. Reitter, "Aquamcs Emacs," David Reitter. http://aquamacs.org/, 2007.

[25] "All Music Guide," http://www.allmusic.com, 2007.

[26] M. White, M. Steedman, J. Baldridge, and G.-J. Kruijff, "OpenCCG: The OpenNLP CCG Library," http://openccg.sourceforge.net, 2007.

[27] I. Androutsopoulos, J. Oberlander, and V. Karkaletsis, "Source Authoring for Multilingual Generation of Personalised Object Descriptions," *Natural Language Engineering (pre-print),* 2005.
[28] N. Mayo, M. Corley, F. Keller, and F. Jaeger, "WebExp2 Experiment Design Software," Edinburgh, UK: University of Edinburgh, http://www.webexp.info, 2007.
[29] R. Likert, *A technique for the measurement of attitudes*: New York, 1932.
[30] K. R. Scherer and M. R. Zentner, "Emotional effects of music: Production rules," in *Music and emtion: theory and research*, P. N. Juslin and J. A. Sloboda, Eds. Oxford: Oxford University Press, 2001.
[31] N. Mayo, M. Corley, and F. Keller, "WebExp2 Experimenter's Manual," Edinburgh: University of Edinburgh, 2005.
[32] "Language Experiments: The Portal for Psychological Experiments on Language," WebExp Development Team, http://www. language-experiments.org, 2007.
[33] "Gumtree," http://www.gumtree.com, 2007.
[34] J. H. Krantz, "Psychological Research on the Net," Hanover College, http://psych.hanover.edu/research/exponnet.html, 2007.
[35] "Amazon.co.uk," Amazon.com, Inc., http://www.amazon.co.uk, 2007.
[36] O. Sessink, "Bluefish Editor," OpenOffice, http://bluefish.openoffice.nl/, 2007.
[37] "SPSS 15.0," SPSS, Inc. http://www.spss.com.
[38] G. E. Dallal, "How to Read the Output From One-Way ANOVA Analyses." vol. 2007: Tufts University, http://www.tufts.edu/~gdallal/aov1out.htm, 2007.
[39] D. E. Rumelhart and D. A. Norman, "Analogical Processes in Learning," in *Cognitive Skills and Their Acquisition*, J. R. Anderson, Ed.: Lawrence Erlbaum Associates, 1981.
[40] M. Milosavljevic, "Augmenting the User's Knowledge via Comparison," in *Sixth International Conference on User Modeling*: CISM, 1997.
[41] M. Milosavljevic, "Introducing new concepts via comparison: A new look at user modeling in text generation.," in *Fifth International Conference on User Modeling, Doctoral Consortium*, 1996, pp. 228-230.
[42] I. Zukerman and R. McConcachy, "Generating concise discourse that addresses a user's inferences," in *Thirteenth International Joint Conference on Artificial Intelligence* Chambery, France, 1993.
[43] K. F. McCoy, "Generating context-sensitive responses to object-related misconceptions," *Artificial Intelligence,* pp. 157-195, 1995.
[44] "All Music Guide," All Media Guide, http://www.allmusic.com, 2007.
[45] G. Carenini and J. Moore, " Generating and Evaluating Evaluative Arguments," *Artificial Intelligence,* vol. 170, pp. 925-952, 2006.
[46] V. Demberg and J. Moore, "Information Presentation in Spoken Dialogue Systems," in *12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2006.
[47] J. Moore, M. E. Foster, O. Lemon, and M. White, "Generating Tailored, Comparative Descriptions in Spoken Dialogue," in *Seventeenth International Florida Artificial Intelligence Research Society Conference*: AAAI Press, 2004.
[48] M. Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy, " Generation and Evaluation of User Tailored Responses in Multimodal Dialogue," *Cognitive Science,* vol. 28, pp. 811-840, 2004.
[49] F. Keller, "Experiment on Sentence Judgments." vol. 2007: University of Edinburgh, http://fordyce.inf.ed.ac.uk/users/keller/res_english4.instr.html, 2007.
[50] A. N. Karasimos, "Evaluation of M-PIRO system text output," in *School of Philosophy, Psychology, and Language Sciences*. vol. MSc: University of Edinburgh, 2003.
[51] "Statistics Glossary: Hypothesis Testing." vol. 2007 Lancaster: Centre for Applied Statistcs Lancaster University, http://www.cas.lancs.ac.uk/glossary_v1.1/hyptest.html, 2007.