Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Venkatesh Bharadwaj

Entitled
Aural Mapping of STEM Concepts Using Literature Mining

For the degree of Master of Science

Is approved by the final examining committee:

Mathew Palakal
Chair

Rajeev Raje

Yuni Xia

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Mathew Palakal

Approved by: Shiaofen Fang 7/13/2012
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

Aural Mapping of STEM Concepts Using Literature Mining

For the degree of Master of Science

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Venkatesh Bharadwaj

Printed Name and Signature of Candidate

7/13/2012

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

AURAL MAPPING OF STEM CONTENTS USING LITERATURE MINING

A Thesis

Submitted to the Faculty

of

Purdue University

by

Venkatesh Bharadwaj

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2012

Purdue University

Indianapolis, Indiana

This work is dedicated to my family and friends.

ACKNOWLEDGMENTS

I am heartily thankful to my supervisors, Dr. Mathew Palakal and Prof. Steve Mannheimer, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

I want to thank Dr. Rajeev Raje and Dr. Yuni Xia for agreeing to be a part of my Thesis Committee.

I would extend my thank Ms. Meelia Palakal for creating the gold standard data used in this work.

Thank you to all my friends and well-wishers for their good wishes and support. And most importantly, I would like to thank my family for their unconditional love and support.

TABLE OF CONTENTS

Page

LIST OF TABLES . . . vi

LIST OF FIGURES . . . vii

ABSTRACT . . . viii

1 INTRODUCTION . . . 1
- 1.1 Overview . . . 4
  - 1.1.1 Classification of words . . . 5
  - 1.1.2 Generating Sounds . . . 6
  - 1.1.3 Generation of sound combination . . . 7

2 RELATED AND PREVIOUS WORK . . . 10
- 2.1 Audemes and their Implementation in Pedagogy . . . 10
- 2.2 Translation of text to aural information . . . 15

3 METHODOLOGY . . . 20
- 3.1 Overview of Methodology . . . 20
- 3.2 Sentence Extraction . . . 23
- 3.3 Phase-1: Word Sequence Generation . . . 24
  - 3.3.1 Need for Classification . . . 24
  - 3.3.2 Classifier . . . 27
  - 3.3.3 Stanford Dependency Parser (SDP) . . . 28
  - 3.3.4 Implementation of SDP . . . 31
  - 3.3.5 The Classifier . . . 35
- 3.4 Phase-2: List of Atomic-Sound Generation . . . 40
  - 3.4.1 Removal of stop-words . . . 41
  - 3.4.2 Synonyms from Online Thesaurus . . . 43
  - 3.4.3 Synonyms from WordNet . . . 44
  - 3.4.4 Sound-word database . . . 45
  - 3.4.5 Weightage and Ranking . . . 46
- 3.5 Phase-3: Audeme Generation . . . 49
  - 3.5.1 Correlation Factor . . . 50
  - 3.5.2 Updating Correlation Factor . . . 51

4 RESULTS . . . 55
- 4.1 Phase-1 . . . 56
  - 4.1.1 Manual Classification . . . 56

Page

4.1.2 Classifier performance 58
4.2 Phase-2 63
4.2.1 Fetching Synonym 63
4.3 Phase-3 66
5 CONCLUSION and FUTURE WORK 69
5.1 Future Work 69
5.2 Conclusion 70
LIST OF REFERENCES 72
APPENDIX 77

LIST OF TABLES

Table		Page
3.1	Two-word prepositions Stanford Dependency Parser can Collapse	32
3.2	Dependency list for definition of "*digestion*"	34
3.3	Classification of words in definition of "*digestion*"	39
3.4	List of words for definition of "*digestion*" after stop-word removal	41
3.5	Synonyms from online thesaurus for the word "*process*"	44
3.6	Synonyms extracted from WordNet for the word "*process*"	45
3.7	Phase-2 output for "*digestion*"	48
3.8	Phase-2 output for "*precipitation*"	49
3.9	Atomic-sounds with correlation factor (integer separated by ':')	51
3.10	Atomic-sounds with updated correlation factors and ranking	53
4.1	Manual classification of a simple sentence describing '*Nebula*'	57
4.2	Manual classification of a complex sentence describing '*Nebula*'	58
4.3	A subset of the transitions extracted from manual classification	59
4.4	Comparison of manual and rules based classification	60
4.5	Sequence comparison between manual and rules based classification	61
4.6	Dependency list for a sentence describing "*precipitation*"	62
4.7	Snippet of sound-word database for atomic-sounds selection	65

LIST OF FIGURES

Figure Page

3.1 Overview of functionality for automatic audeme generation 22

3.2 A typed dependency parse for "I saw the man who loves you" 31

3.3 Phrase tree structure of a sample sentence 33

3.4 Dependency graph generated using grammar scope 35

3.5 Automaton for rule based classifier, with a subset of rules 38

3.6 Overview of processing of Phase-1 with an example of "*Precipitation*" . 39

3.7 Processing for Phase-2 42

3.8 XML output generated by thesaurus.com API 43

ABSTRACT

Bharadwaj, Venkatesh. M.S., Purdue University, August 2012. Aural Mapping of STEM Contents Using Literature Mining. Major Professor: Mathew Palakal.

Recent technological applications have made the life of people too much dependent on Science, Technology, Engineering, and Mathematics (STEM) and its applications. Understanding basic level science is a must in order to use and contribute to this technological revolution. Science education in middle and high school levels however depends heavily on visual representations such as models, diagrams, figures, animations and presentations etc. This leaves visually impaired students with very few options to learn science and secure a career in STEM related areas. Recent experiments have shown that small aural clues called Audemes are helpful in understanding and memorization of science concepts among visually impaired students. Audemes are non-verbal sound translations of a science concept. In order to facilitate science concepts as Audemes, for visually impaired students, this thesis presents an automatic system for audeme generation from STEM textbooks. This thesis describes the systematic application of multiple Natural Language Processing tools and techniques, such as dependency parser, POS tagger, Information Retrieval algorithm, Semantic mapping of aural words, machine learning etc., to transform the science concept into a combination of atomic-sounds, thus forming an audeme. We present a rule based classification method for all STEM related concepts. This work also presents a novel way of mapping and extracting most related sounds for the words being used in textbook. Additionally, machine learning methods are used in the system to guarantee the customization of output according to a user's perception. The system being presented is robust, scalable, fully automatic and dynamically adaptable for audeme generation.

1 INTRODUCTION

Improving science education has been identified as imperative for all industrialized nations. As technology is increasingly driving every aspect of daily life, people at all levels of business, industry and mainstream society are asked to understand the basic underlying scientific concepts and processes that fuel technological advancement, whether to contribute to this progress or to evaluate its impact on society. The education that leads to such understanding must begin at the elementary level. As pointed out by Peggy Tilgner [1] science education is critical in elementary schools.

More than other traditional classroom subjects, science is a field that benefits from or even requires special pedagogical tools to help students understand a concept. Some of the most frequently used tools in science textbooks and classroom pedagogy are essentially visualizations: models, diagrams, illustrations, charts, photos, as well as videos, etc. The use of visualizations has increased dramatically in the past half-century and today it is hard to imagine science education without them. As noted by Slough et al. [2], imagery has come to play a prominent and at times dominant role in science pedagogy. This is due in part to the greater ease and lower cost of graphic design and color printing technology, the increasing influence of image-heavy Website design, the general proliferation of screen-based pedagogy, and demonstrably positive outcomes. Visualizations can be key to making science content more engaging or interesting and can offer details or comparisons that would otherwise require overly elaborated text explanations, and can even provide the main vehicle for content delivery, with only subordinate text labels [3]. Others agree with the old adage, "a picture is worth a thousand words" (given by Fred R. Barnard [4]), like Hibbing et al. [5]. Lowe [6], said graphics can facilitate comprehension of science concepts where a textual explanation might require lengthy convoluted descriptions. The power of visualization is so great, in fact, that Gilbert et al. [7] considered it essential for the students to grasp science concepts, particularly when subjects are too large, too small, too brief or too remote to be observed first hand. Hestenes [8] also said that science education relies on concept modeling which often requires visualized models.

The dependency of science education on visual tools is increasing at a much faster rate with the advent of affordable and powerful handheld devices. More and more educational content is designed for a wide range of technology platforms from traditional textbooks to e-books to handheld tablets, making science education further more dependent on visual contents like animation and motion graphics. It has been shown in previous studies that using these resources the effective engagement of high school students in science can be improved [9]. Also as proposed by Tytler et al. [10] this engagement is crucial for building long term interest of science. All the above mentioned factors together suggest that mainstream science textbooks and other pedagogy mediums used in middle schools will remain, heavily dependent on visualizations and visual tools.

Although the use of visual tools increases the interest and engagement of student in science, it also presents severe challenges to Blind and Visually Impaired (BVI) students. Because most blind schools aspire to offer high school diplomas to their students, they use state-approved textbooks which are dependent on visualizations. The unintended message for BVI students is that science is for the sighted people only. The BVI community, however, rejects the message that science education is meant for sighted people only. As presented in a concept paper from the National Center for Blind Youth in Science (NCBYS) of the National Federation of the Blind [11], "Historically blind students have received especially inadequate training in science and math concepts, particularly during the critical middle and high school years, when a passion for a subject and career interest is best sparked." Society must "develop new educational products that are based on the true needs of blind students" and also "based on meaningful research." The ideas offered by Bernhard Beck-Winchatz of DePaul University in his address to the 2002 convention of the National Federation of the Blind, "Can Blind People be Astronomers?" Beck-Winchatz makes this fundamental argument for blind scientists: Science today is largely a matter of interpreting data and asking interesting questions about those data and related information. Regardless of the original sensory modality gathering or conveying the data, the "science" begins when someone asks a new question either seeking new data, or suggesting a novel interpretation of the existing data. Success in science education depends not simply on the ability to perceive visualized information but rather on the inductive, deductive, or metaphoric thinking that comes from constructing innovative models through which students understand, interpret and cross-correlate this information.

With this in mind, to emphasize the *conceptual* rather than the *perceptual* processes of science education, and to help remedy the disadvantage suffered by BVI students of science, we propose an automated system for generating non-speech sounds to "aurally illustrate" science concepts. The learning materials being developed can accompany the regular classroom pedagogy activities at school for the BVI or mainstream schools with BVI students. We are also motivated by the realization that, like so many other subjects, science education will increasingly be delivered via e-books and tablet devices with touch screens and multimedia capacity. Designing auditory information to complement text and visualization presents and opens an evolving opportunity. Here we present one approach that promises to make science education not only more engaging for BVI students but also powerfully complements the existing text and vision pedagogies. The methods employed in getting to the results include tasks requiring Natural Language Processing and Information Extraction along with some attributes of machine learning. The aim can also be stated as to find an alternate non-speech aural representation of science concepts. The proposed methods are specifically designed for the domain of science education only.

1.1 Overview

Significant prior research has established the fact that non-speech aural representation can be used to enhance the interest and involvement of BVI students in science and related subjects [12]. These non-speech aural representation are called 'audemes' which are "short, non-verbal sound symbols made up of 2-5 individual sounds lasting 3-7 seconds" as defined by Mannheimer et al. [13]. These sounds are either quoted from the real world as "sound effects" (e.g. a dog barking, a car starting, waves sloshing on the shore), or musical snippets. Thus defined, audemes are, in effect, "molecular" aural collages comprised of individual "atomic" sounds. Atomic-sounds are aurally quoted from a single, readily identifiable source or illustrative of a single scenario (e.g. the sounds of hammering and sawing, although two distinct actions, can be heard simultaneously to illustrate a single unifying concept-as-scenario: "construction"). This thesis presents automated methods which can generate an audeme for a concept by drawing from a pool of atomic sounds.

To find a suitable sound for a given science concept, our system has to first understand, what underlying concepts are being used to describe this concept. Although topics that are taught in grade K-12 level science are basic in their depth of details, they spread across a broad array of subtopics. This makes the task of information extraction a challenge, because there is no prior knowledge base for this type of generic science process/concept definitions. Also, although most sentences in science textbooks are straightforward declarative statements, with some questions or interrogatives, the natural latitude of English permits a wide variety of grammatical and syntactic arrangements. Thus it becomes imperative to use Natural Language Processing techniques in order to enable the system with the required information. Fortunately, the sentences used in a science textbook can be assumed to be grammatically correct. This gives us the opportunity to exploit English grammar for Information Extraction (IE).

1.1.1 Classification of words

Because our goal is to design a system to create audemes to complement the text in a standard science textbook, and given the range of sentences found there, our first problem addressed in this thesis is correctly identifying the sequence of things, actions and modifying conditions that together form the verbal explanation of a science process. The methods described for solving this problem of IE from the description of a concept/process in a textbook involves use of multiple Natural Language Processing tools. IE methods used in this thesis depend on the classification of words according to their role in a process. The roles can be defined as the contribution of an object or action to the overall process. The classification of objects and actions being performed in a process as per their role in that process gives the information about the sequence of events in that process. This classification can also be considered as a generic encapsulation of any attribute used in a science process or assertion. Apart from solving the problem of finding the sequence of events in a process for sound generation, this encapsulation can also be used as a tool for explaining a science process in general. According to our understanding each word used in defining a process can be classified in either of the following five categories or classes of semantic roles: 1)Initiator 2)Condition 3)Action 4)Action-on 5)Output. Further details on these classes and their meaning are discussed in Chapter 3. These five semantic roles are further referred as classes in this thesis.

This classification of major words is done for every sentence in the textbook that is used to explain the concept at hand. These sentences are fetched from the textbook in OCR (Optical Character Recognition) format. In order to classify the words in a sentence, grammatical dependencies among related word pairs in a sentence are extracted using the Stanford Dependency Parser (SDP) [14]. This parser takes into account the Part of Speech (POS) tagging of words and creates a parse tree of a sentence before allocating a typed-dependency to a pair of words. The dependencies generated by SDP are then read by a rule based classifier that allocates those words to one of the five classes, according to the dependency that drives that word. The arrangement of words in different classes gives the sequence of concepts (events/actions, objects and conditions) we need for our analysis.

1.1.2 Generating Sounds

The second problem addressed by this thesis is to generate a list of words that are synonyms of the base word in any given class of the process, with particular attention focused on any word that readily correlates to an aural event or sound effect. For example, water in a natural spring may bubble and bubbling immediately suggests an aural event. Solving this problem is required due to the fact that certain words are best suited to explain a thing or action verbally but they may not have an obvious sound effect associated with them. The classification that has already been done in already discussed NLP steps needs to be translated as a sequence of sounds. For this to happen we require sounds that correspond to the words in a class as well as portray the same actions, things or modifiers as the word itself would. In order to fetch the synonyms for a base word we used WordNet [15] and thesaurus.com (which is based on the *Roget's Thesaurus*) [16] as external lexical resources. Both the resources are independent of each other so both are given equal weightage in the ranking of synonyms based on their relevance. The relevance of a synonym is calculated on the basis of relative frequency of their occurrence in synonym list for a class in definition of a concept.

$$RelativeFrequency = \frac{number\ of\ occurrences\ of\ a\ synonym\ in\ a\ class}{total\ number\ of\ synonyms\ in\ a\ class} \tag{1.1}$$

Other metrics used to increase the ranking of any synonym are: a) its occurrence in a manually created database of preferred sound-words (words that possess an intuitive or onomatopoetic aural equivalent, e.g. bubble or buzz); and b) if the synonym is a verb or a noun/verb (e.g. hurry, fall, drop, splash). In the sound database, these sound-words are directly linked to a corresponding atomic-sound. These associations are created manually done by giving word tags to each atomic-sound in the database.

1.1.3 Generation of sound combination

After creating the ranked synonym list, our system generates a list of atomic-sounds, i.e. the sounds associated with a single “thing” (object or action). A single thing like water can have multiple sound signifiers/corollaries associated with it depending on the conditions. For example: *water_stream*, *rain_fall*, *water_drinking*, *water_splash*. All of these are sounds of water but in a certain context it makes more sense to choose one sound over other. This decision to select a single sound file is subtle and people may reasonably disagree as to the most appropriate choice. Our system required a scheme that would aggregate the data generated by the play of audeme games, then dynamically change the result of selecting one sound out of the list of possible audemes (sound signifier/corollaries). This is done by taking feedback of the games being played at the Indiana School for Blind and Visually Impaired (ISBVI). The data collected helps in ranking the sounds for a particular concept and this is done by taking into account the count of number of times a single option (out of the multiple options for sound signifiers presented in the game) is selected by a number of students. This count is further used to increase the ranking of a single atomic-sound in the list of atomic sounds. The change in this ranking dynamically changes the audeme by picking up the latest top ranked atomic-sound to form audeme.

Use of audemes has already been proven to be beneficial as a learning medium for BVI students [17]. In order to make games and other multimedia applications based on audemes, we need both an adequate supply of audemes and a process for automatic generation of audemes. At present, the creation of audemes is a manual task.

Moreover, this process is inherently based on, and thus limited to the imagination and perceptions of the individual audeme curator/creator, whose choices for sound-concept associations may differ from the majority of BVI student's perceptions. With the new techniques presented in this thesis we can have an automated system to generate audemes at a much faster rate and provide multiple options for an audeme signifying a science concept. The feedback mechanism provided by the games played online or via computer or other handheld devices can further improve the semantic relevance of the audemes generated for any given concept.

The methods presented in this thesis, are best applied to the generation of the "micro-narrative" type of audemes, as opposed to the "metaphoric" type. As described by Back M. et al. [18] micro-narrative is a sound design technique where "The sound designer does not attempt to replicate 'real' sounds; the task is rather to create the impression of a real sound in a listener's mind. In this attempt to create a sound in the listener's mind, the sound designer is aided by user expectations based upon cultural experience as well as physical experience".

In our work, micro-narrative audemes use such "real" or virtually real sounds in an intuitively obvious sequence that generally suggests a cause-and-effect relationship or narrative unifying the aural components: e.g. the sounds of a barking dog, then a baseball bat hitting a ball, followed by the crash of breaking glass tells a micro-narrative of a children's game gone awry. The other main type of audeme is metaphoric, in which the sounds have either a thematic connection to the verbal content (e.g. a fire siren signifying fire/heat + a yodel signifying mountainous terrain = fire/mountain = volcano) or even a punning connection (e.g. the sound of coins spilling on a table = change = transformation, then combined with the sound of rock music = rock to concatenate as "transformed rock" = "metamorphic rock"). Metaphoric audemes often rely on a pre-stated domain to help listeners decode their significance. Although the methods presented in this thesis, are aimed at micro-narrative generation but still the system does not guaranteed the exclusion of metaphoric references to any action, modifier or a thing.

2 RELATED AND PREVIOUS WORK

This chapter is divided into two main sections. The first section discusses previous studies about the generation, effectiveness and use of audemes and aural representation as a medium of pedagogy for BVI students. The second section discusses the work that has already been done to generate the tools used in our system for automatic audeme generation.

2.1 Audemes and their Implementation in Pedagogy

Non-speech sounds have been used to signify information or to serve as user notifications in computer interfaces since 1980s. The two most common categories of non-speech sounds are earcons (generally abstract sounds such as beeps or tones) and auditory icons (generally imitating some real-world sound appropriate to the signified function). Earcons are very common and are used to grab the user's attention, in computer operating systems. Earcons are defined as "brief, distinctive sound used to represent a specific event or convey other information." [19]. Brewster [20] investigated the use of earcons as a means of presenting information in sound and proposed from his experiments that parallel earcons can increase sound presentation rates. He also proposed that non-speech sounds could be used to overcome the problems of conveying hidden information. Buxton [21], proposed that ability to receive information from non-speech sounds has a potential to be used as an aid in helping improve the quality of human interaction with complex systems.

Research on auditory stimuli investigates the ease of learning when using sound effects (SFX) to enhance retention of educational content and improve academic outcomes.

Stephan et al. [22] investigated how pre-existing association between sound and content influences the ease of learning and retention when pairing auditory icons with warning events. In that study each auditory icon was classified as direct, related, or unrelated in comparison to the warning event. For example, a sound snippet of a dog barking had a direct relationship with the word dog, a related relationship with cat, and an unrelated relationship with waves. For the study 63 participants were randomly assigned to one of the three conditions for each of the 24 auditory icon pairings. Participants took part first in an initial session where icon-word pairings were learned and a follow-up session four weeks later to test retention of these pairings. Participants sat at a computer and listened through headphones to each auditory icon as it was paired with the printed word on screen. Participants had better learning and retention outcomes with auditory icons that had a direct or related association with its referent. Similarly, Keller and Stevens [23] paired auditory icons with pictures and words that either had a direct relationship or were indirectly related ecologically or metaphorically. For example, the image of a helicopter, presented with the text HELICOPTER, had a direct relationship with a helicopter sound, an indirect ecological relationship with gunfire, and an indirect metaphorical relationship with mosquito. With 90 participants in the first experiment, a one-way ANOVA revealed that indirect association strength was greater for ecological conditions than metaphorical. In experiment two, 64 participants were assigned to each condition before given nine training and test trials. Researchers predicted it would take the fewest number of trials for participants to learn direct relationships, followed by ecological relationships and then metaphorical relationships. Direct, meaningful relationships between the sounds and the target referent were quickly learned by participants. There was not a significant difference between the number of trials it took to learn ecological and metaphorical conditions. However, associations with indirect relationships were quickly recognized after participants had been exposed to the sounds.

There has been a significant amount of research into the impact of music on human thought processes. Koelsch et al. [24] summarizes the four commonly discussed types of semantic meaning that music can evoke:

- Meaning that arises from common patterns of sound (e.g. flutes imitating bird calls, etc.).
- Particular emotional moods.
- Meanings from external associations such as a national anthem or perhaps very low tones used to evoke a low physical location.
- Meanings that arise when musical passages parallel narrative structures or sequences of tension-then-resolution.

This study explores the well-known phenomenon of semantic priming in which a pervious stimulus prepares the listener to more readily process or understand a new stimulus as contextually related. They speculated that passages of a Beethoven symphony would be much more likely to semantically prime the word "hero" rather than the word "flea", and their experimental work demonstrated the validity of this hypothesis. In other words, music that engages at least one of the four modes of meaning can contribute to the semantic processing and conceptual associations brought by the listener to subsequent words. This power of semantic priming is fundamental to understanding audemes, and how a few seconds of SFX or brief snippets of melody can semantically prime the listener to understand subsequent sounds in loosely focused and non-arbitrary contexts.

The rapid evolution of audio technologies has expanded our understanding of the overall role and potential application of sound in culture. This has prompted scholars such as Erlmann [25] to reassess vision-centric theories of sight and text as the primary vehicles of cognition, education and cultural knowledge. It has also catalyzed the development of more practical adaptive technologies (e.g. screen-readers, refreshable Braille displays, etc.) to help communicate visual information to BVI students as well as sonification or audification strategies to aurally illustrate quantitative data through abstract tones. Starting in the late 1980s, important work in the use of aural cues for computer interfaces was performed by Edwards [26] and Brewster [20]. Back and Des [18] also indicated that listeners expect the natural world to sound like the SFX in popular media, and infer micro-narratives or brief scenarios from these SFX.

Previous work done by Mannheimer S and colleagues at Indiana University-Purdue University, Indianapolis (IUPUI) [13, 17] in partnership with the Indiana School for Blind and Visually Impaired (ISBVI) proves that use of short non-speech sound collages associated with educational content can significantly improve the recall of that content. The group has proposed the term "audeme" (suggesting a general similarity to terms such as morpheme, lexeme and/or phoneme) to mean a combination of "atomic" sounds from a single auditory source or event (e.g. surf on the shore, dog barking, a descending musical scale, etc.) crafted into a brief molecular audio collage. Audemes generally last 3-7 second, and are generally used to signify a specific educational topic or theme (e.g. igneous rocks, the water cycle, the American Civil War) and to prompt memory of an associated body of verbal content. Audemes may combine 1) iconic sounds made by natural and/or manufactured things (e.g. surf and seagulls, barking dogs, hammering nails); 2) abstract sounds generated by computers (e.g. buzzes, blips, etc.); and 3) music. Experiments at the ISBVI demonstrated that audemes can serve an effective alternative to visual and textual labels/icons for verbally presented educational material. They determined by a series of experiments that a combination of 2-5 separate atomic sounds works best.

The group performed experiments where three separate groups of students were taught the same essays about "Slavery", "Radio" and "US Constitution". Three audemes, each representing one of these themes, were prepared by researchers. Group

1 never heard the audemes; group 2 heard the audemes during the learning phase; and group 3 heard the audemes during the learning and testing phases. After fifteen days all three groups were evaluated for their memory through a test. During the test groups 1 and 2 were not exposed audemes, but group 3 was exposed to the appropriate audemes. The results of the experiment confirmed that exposure to audemes significantly increased in recall of groups which were exposed to them during the learning and/or testing phases, with the greatest increase coming in group 3, which heard audemes during both phases. Related experiments also suggested that audemes which were thematically or metaphorically related to the target themes were more effective than audemes that simply presented unusual sounds; and also that there was audemes judged by students as displaying positive affect were more useful as memory prompts than audemes with a negative affective quality.

Similar research by Gaver [27] also suggested that iconic sounds have a better impact as memory cues for long-term memory and a deeper understanding of content associated. Sanchez [28] also proposed the effectiveness of sound based computer interfaces to enhance educational understanding in BVI children. The work by Doucet et al. [29] suggested that blind people have a better performance in processing auditory information as compared to their counterparts with sight. One reason for this can be that BVI people use their acoustic senses more than sighted people, thus enhancing their aural capabilities [30].

The work done by Ferati et al. [31]; has been informed by this foundation and also parallels other work targeting commercial markets (Roma et al. [32]). The methods presented in this thesis, relies on semantic flexibility to allow audemes to demonstrate various ecological or metaphoric aspects depending on the context. The discussion of ecological and metaphoric sound (Keller and Stevens) [23] is valuable in understanding the general concept of audemes in our work, although the work presented in this thesis takes the idea a step farther in its use of complex concatenations of sounds and contextual meanings. In the concept of audeme sequences presented here, we propose that because audemes can signify different (although semantically related) concepts in different contexts, listeners must actively construct these concepts on a case-by-case basis rather than simply recognizing the audeme as a mnemonic cue for a single pre-established concept. For this reason, our database of atomic-sounds also functions as a semantic thesaurus to suggest a range of audemes and audeme-sequences that might reasonably signify the same or similar concepts.

2.2 Translation of text to aural information

The audemes used by Mannheimer S. et al. [13] for their experiments at ISBVI were designed and created by the researchers themselves, with consultation from the student users. Although this manual process is useful and engaging, the long-term strategy for audemes in education is best served by a more broadly-based process for audeme creation, particularly through an automated process that facilitates audeme creation by many different user groups at multiple sites, and also aggregates data from both the creation and use of audemes in various educational games. The aim of this thesis is to design a method to automatically generate audemes based on analyses of the textbook language surrounding target educational concepts in high school science. Automatic generation will ensure the supply of audemes for varying concepts. As previously defined [13], audemes are aural representations of concepts which are semantically equivalent to the text required to explain that concept. Because of this semantic relationship between text and audeme, we have designed a workflow to process text from textbooks used at for the ISBVI, and generate audemes by extracting semantic information present in that text.

Correlation of science concepts from textbooks to audemes involves multiple text-mining and Natural Language Processing (NLP) techniques. This is because textbooks strive for an engaging written style, which calls for a variety of sentence structures, often using several different sentence types (including statements, questions, with active and passive verbs, multiple dependent clauses, etc.) in a broad explanation of a single science concept. Thomas M. [33] found that there are over 14 different ways to express the same relation between objects in a sentence. This adds the ambiguity which needs to be resolved by NLP methods for semantic Information Extraction which can be later translated into aural form.

In this thesis, a major part of the text-mining work concerns semantic Information Extraction of science concepts from the textbook. Since 1990s a lot of research has been done in the field of IE, this is mostly for intelligent analysis of data over the internet, either by financial services companies to seeking information about recent business trends or by search engines for generating search results [34]. A range of tools have been developed for IE over the web. Chang [35] and Laender [36] present a survey of these tools. Another survey of information extraction research has been presented by Sarawgi S. [37] wherein different extraction tasks and techniques used for IE are discussed. The tasks and techniques performed for IE are customized for type of data and the information required from it. In this case we aim at generating sequence of events from the text of textbook explaining a science concept. Some of the most common tasks performed in NLP for Information Extraction, which are also being used in the work presented in this thesis, are: text selection, Removal of stop words, tokenization, Part of Speech tagging, dependency extraction and stemming.

- **Text Selection**: In order to extract relevant text segments from the textbook for further processing, first of all we must convert the textbook into machine readable form. This can be done by performing Optical Character Recognition (OCR) on the text book [38] scans. Relevant sentences are then extracted from the text based on regular expression matching, this is done using 'grep' tool in Linux [39]. The final set of text on which further processing is done for IE contains the description of the science concept at hand.

- **Stop-word removal**: One of the most commonly used technique in any NLP work is stop-words removal. Stop-words are language specific and a lot of research has been done recently in the field of stop-word removal for different languages [40–42]. We have used WordNet [43] stop-words list for stop-word removal from the sentences fetched from textbook. Below is a list of stop-words of English as given in Natural Language Toolkit being used in this thesis.

  Stop words for English extracted from WordNet are:

  i, me, my, myself, we, our, ours, ourselves, you, your, yours, yourself, yourselves, he, him, his, himself, she, her, hers, herself, it, its, itself, they, them, their, theirs, themselves, what, which, who, whom, this, that, these, those, am, is, are, was, were, be, been, being, have, has, had, having, do, does, did, doing, a, an, the, and, but, if, or, because, as, until, while, of, at, by, for, with, about, against, between, into, through, during, before, after, above, below, to, from, up, down, in, out, on, off, over, under, again, further, then, once, here, there, when, where, why, how, all, any, both, each, few, more, most, other, some, such, no, nor, not, only, own, same, so, than, too, very, s, t, can, will, just, don, should, now

- **Tokenization**: Tokenization is one of the most important and an initial phase of NLP. It is identification of tokens or basic units which need not be decomposed further for processing [44]. There can be multiple ways of doing tokenization depending on the requirement. For example text can be tokenized by using regular expressions (which can be used as a tool in itself) [45] or by using custom build tools such as word_tokenize() function provided in Natural Language Toolkit in python [46]. This function extracts individual words from a sentence as tokens, and generates a list of those tokens.

- **Part-of-Speech Tagging (POS tagging or POST)**: Part of speech tagging is another field of research in itself and it is also an important part of NLP. As the name suggests POS tagging assigns a tag to each word which corresponds to its part of speech as used in the sentence. Multiple POS taggers have been developed which are either rule based [47], maximum entropy framework using n-gram extracted from tagged corpora [48], based on hidden Markov model [49] etc. Stanford tagger [50] is another tagger which uses a log-linear model for finding POS tags, it considers both preceding and following tag contexts using a bidirectional dependency network for finding the tags.

- **Dependency Parser**: A dependency parser works over the sentence to read the grammatical structure of sentences [14] or syntactic and semantic structure [51, 52]. It identifies the groups of words which fall together along subject and objects of a verb depending on sentence structure. Most of these parsers either take probabilistic or statistical approach.

- **Stemming**: Stemming is one of the most simple and common technique used in Information Retrieval to ensure correct matching of morphologically related words. It is used to reduce the inflections of a word to their common root. Most of the time this is done by removing suffixes from the word like '*ing*', '*tion*', '*es*' etc. There are many algorithms in use for stemming [53]. One of the most popular stemmer is Porter's Stemmer [54] which has been used in the methods presented in this thesis.

There are numerous NLP tools and resources present for use these days, each having their own qualities and drawbacks. We are using only a few of those tools as per our requirements. One of the most important resources required in overall work of audemes generation from text are the resources for synonym list generation. There are many lexical resources which provide a list of synonyms, a majority of them use either WordNet [15] or Roget's Thesaurus [55] as their primary source and knowledge base. There has been a lot of research being done to extract synonyms from both the above mentioned resources [56, 57]. In recent days many online dictionaries and thesaurus have been generated and have become major medium of using these resources [58].

Since Natural Language Processing is the base of the work presented in this thesis and one of the method almost every NLP tool needs is Machine Learning (ML). Machine learning means enabling a system with some knowledge which is learned from the training data provided to the system and based on that knowledge, decisions are taken for further actions. There have been a lot of work done on machine learning and several algorithms have been designed depending on the requirements and type of data being used. Some of the types of learning machine learning methods that are commonly used are Supervised Learning, Unsupervised Learning, Reinforcement Learning and Evolutionary Learning [59]. Different ML techniques may fall in between these types. Much of the work in Data-Mining and Artificial Intelligence relies on some form of learning algorithm. Since we are dealing with the system that operates using multiple NLP techniques we are actually using this learned data. For example POS tagger, dependency parser depends on Supervised Learning. We have also presented, in this thesis, a new learning algorithm for phase-3 of processing wherein the system takes into account the feedback from users about their perception of best audeme for a science concept. The system uses this method for making changes in the sound selected out of a list of atomic-sounds, thus adapting to the user feedback and their perception.

3 METHODOLOGY

3.1 Overview of Methodology

In order to automatically generate audemes for a science concept or process we have to first analyze the output required, that is the audeme itself. By definition audeme is a sequential combination of multiple individual sound snippets called atomic-sounds, where each atomic-sound corresponds to a single action, thing or modifier of a process. One example of atomic-sound is a sound of *fire-siren*, which may portray heat or fire. A peculiar and notable aspect of any science process is the sequence of events that take place as attributes of process. If the sequence of events is changed in the definition of a science concept, then the definition of same concept may not make sense. For example the definition of *digestion* as given in the glossary of Glencoe Blue book is: *"chemical and mechanical process that breaks food down into small molecules so that they can be absorbed by the body."* In this definition the sequence of attributes is: 1) "chemical and mechanical process". 2) "break food into small molecules". 3) "absorbed by body". Similarly for the definition of *hydroelectric power*, which is "*electricity produced when the energy of falling water turns the blades of a generator turbine."*, the sequence of events is 1) falling water energy 2) turns blades of generator turbine 3) electricity produced. In both the above examples we can analyze that if the sequence of events is changed then it would be hard to explain the same process. The same will also hold for an audeme, since they are supposed to be aural translations of a science concept. That is, an audeme can portray a process correctly if the sequence of individual atomic-sounds corresponds to the sequence of events in the process as explained in the text definition of a process. This semantic sequence should not be confused by the sequence of words, instead this is the sequence of events that occurs in a process. The mapping of text to audeme here is more semantic rather than syntactic. Methods for audeme creation presented in this thesis are designed only for aural micro-narratives generation of a science process. Micro-narrative, describe a sequence of events, so we have to have a sequence of sounds or words for audeme generation. The input to this system is a textbook and as already stated in the previous chapter that a single sentence can be written in different formats, that is, two sentences defining the same process may have different arrangement of words within a sentence. As presented by Thomas M. [33] a single sentence can be written in 14 different structural variations. The simplest example would be active and passive sentences in English that have different arrangement of object and subject around a verb. In order to maintain the semantic sequence we decided to classify, the major words (ignoring stopwords) used in sentences that define the process into five categories. These classes can then be arranged sequentially to produce the same sequence of events. This classification of words for sequence generation is the first Phase of overall processing done for automatic generation of audemes.

The second Phase of processing for automatic generation of audemes is, generation of a list of sounds associated with the words extracted in Phase-1. This Phase is the transition Phase from words to atomic-sound. In this Phase the system collects related words (synonyms) from external resources. This is required due to the fact that the words used for explaining a science process in a textbook definition may not have a direct sound affect associated with them. In order to find a correlation between a concept at hand and the sound that can best represent it, we need words whose sounds are already known to the system and still represent the same concept. These words, also called sound-words, are fetched by searching known sound words in the list of synonyms generated for a base word. The base word here is a single word, which was extracted and placed in one of the five classes in Phase-1.

The processing for third and the final Phase of audeme generation contains selection of a single atomic-sound represent an event out of multiple possible atomic-sounds for it. This is required due to the fact that a single action, thing or modifier which is part of a process can be portrayed as different non-verbal sounds depending on the situations in which it is present. In order to create best possible audeme we have to pick up the sound which best correlates to the process and the conditions in which the event/object is present. For example in the above definition of *hydropower energy* the term water is used which is an integral part of the representation of hydropower energy, but water can be presented by many types of sounds. For example *rainfall, water-drop, water-gulp, water-splash, water-stream, water-fall* etc. In the audeme for *hydropower energy* it makes more sense to use either *water-stream* or *water-fall* than other sounds of water, to portray water aurally.

Thus we saw that the operations being performed for generation of Audemes from textbook can be broadly classified into three phases, as shown in Figure 3.1.

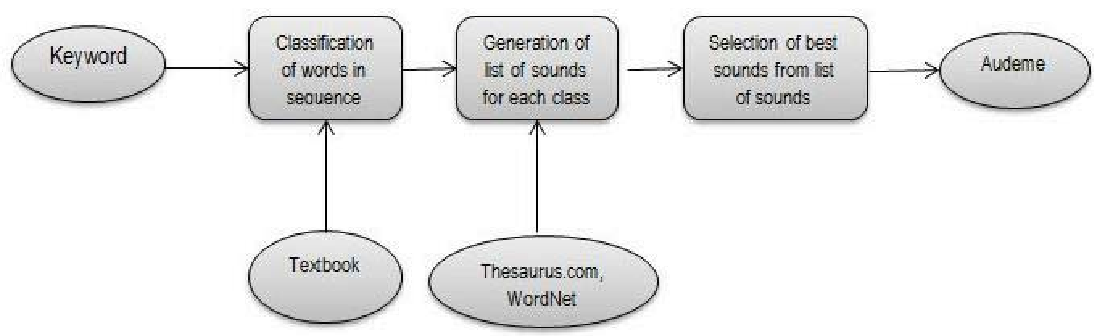

Figure 3.1. Overview of functionality for automatic audeme generation.

The following sections describe in detail the processing being done for automatic generation of audemes in each Phase. But before starting with first Phase of processing let us take a look at preprocessing steps for input data creation.

3.2 Sentence Extraction

Before starting the processing for audeme generation data needs to be prepared, which can be used as input to the system. The input or the starting resource in this process is text from textbook used in middle school. We took Glencoe Blue book as our initial source of science concepts. The book was translated into computer readable text format after scanning it and running Optical Character Recognition on the scans. Since OCRs are not yet 100% accurate and given that the science book contains a lot of diagrams which further confuses OCR, we had to do some manual formatting of text. Once we had the text file in proper format, then for every science concept under consideration, multiple sentences, which are used to describe the science concept, were extracted from the text. This is done by simply finding the concerned word in the glossary part of the text and extracting sentence from it. Since the glossary only contains one sentence along with the reference of explanation of the concept in corresponding chapters of textbook, so this reference is further used to read those chapters. Sentences used in the chapters that suggest the concept are extracted and stored along with glossary definition. Word matching can be done using a simple string matching function present in almost any standard programming language. Also alternatively a *grep* like utility can be used for sentence detection and fetching from raw text of a textbook. The sentences fetched are stored as "." delimited in a database for further processing. "." is generally used as sentence boundary in English and many NLP tools are capable to identify it as sentence boundaries.

In current implementation only the sentences containing keyword (name of science process/concept) are taken into consideration, however some selective adjacent sentences can also be taken into consideration for processing. But this comes with an added penalty of fetching a lot of sentences and that leads to ambiguous results because of a large number of words. This completes the preprocessing of data; we now look at the detailed description of multiple processes done on this data for audeme generation.

3.3 Phase-1: Word Sequence Generation

This Phase creates a semantic sequence of words from multiple sentences which can together, when taken in a sequence represent a science process. This sequence is formed by arranging the words in five classes that are explicitly defined for STEM concepts.

3.3.1 Need for Classification

As already stated the semantic sequence of events is one of the most important ingredients in preparing audemes for a science process/concept. Sequence ensures that audemes actually represent the concept for which they are created. In order to create this sequence we have proposed a novel classification of various attributes of a science process/concept. Since this sequence is the sequence of events or conditions that define a process and is not dependent on the structure of sentence, therefore the classes being defined here is generic for all the science processes/concepts.

All action, thing and modifiers words that are used in the sentences describing a science process can be classified in either of the five sections or classes: *Initiator*, *Conditions*, *Action*, *Action on* and *Output*. All these words that are used in a sentence describing a process can be accommodated semantically in one of these classes. Following section discuss the details of these classes:

1. **Initiator**: The understanding behind this class is that every science process has some initiation. This means, there is some event or subjects which triggers a process or is the main actor in the process. This initiator can also be the main subject on which a process is happening. For example in the definition of *'hydroelectric power'* as given in section 3.1, *'water'* is the reason whose falling action causes blades of a generator turbine to rotate, which later produces electricity. So water is acting as initiator in this definition of hydroelectric power. Similarly for the process of *'digestion'*, as per the given definition in section 3.1 *'chemical and mechanical process'* are initiators. Initiators are generally noun words.

2. **Condition**: This class contains all the conditions in which a subject is present or we can say that these are the words that tell about the conditions in which an action is being performed or conditions which are affecting an action. For example the definition of *'petroleum'* as given in the Glencoe blue book glossary is "*nonrenewable resource formed over hundreds of millions of years mostly from the remains of microscopic marine organisms buried in Earth's crust.*" In this definition different conditions that define petroleum are *'hundreds of millions of years'* and *'Earth's crust'*. Conditions can be a place, time or position which is used to explain a science concept. Conditions are also generally noun phrases or noun modifiers.

3. **Action**: Action words are the driving words in a process definition. Since all science processes are some kind of actions in happening and many objects and conditions in a definition of a science process are around an action, this makes the words suggesting an action of utmost importance. Moreover it has been found that most of the sound-words are actually some form of actions. This is also as per the general intuition that it is actions that have a sound associated

with them rather than the objects. A simple example to verify this assertion is the sound of a cup. There is no direct non-verbal sound of a cup but it is the action being performed on a cup that gives it a sound, which can be a *flick* of a finger on the cup, or *gulping* coffee from a cup, or keeping a cup on a table with a *thud*. All the three sounds can be related to the cup but they are actually sound of some action being performed on a cup. Therefore we can see that the action words that are present in a definition of a science process can have a direct correspondence to a non-verbal sound. The words assigned to class *'action'* are most likely to be verbs, since by definition verbs are supposed to be actions. Action words also play a central role among all the five classes, it joins the elements in initiator/condition with the one in action-on/output classes (Action-on and output classes are described below) this can be directly correlated to a subject and object around a verb in a simple sentence. Examples of action words in the sample definition of *'hydropower energy'* given in section 3.1 are: *'produced'* and *'turns'*. Similarly for the definition of *'digestion'* the action words are *'breaks'* and *'absorbed'*.

4. **Action-on**: Action-on contains the words that suggest the objects on which an action is being performed in a science process. Action-on words are directly related to actions. These are also noun words. In complex sentences explaining a science process, since there are multiple actions and many of them occur in a sequence, in those cases action-on gives the list of words on which action is performed. For example in the definition sentence for *'hydropower energy'* given in section 3.1 the *'action-on'* words are *'blades'* (for action *'turns'*) and *'turbine generator'* (for action *'produced'*).

5. **Output**: The last class in the sequence of events is the class of output, as the name suggests it corresponds to the end result of a science process. Almost every science process has some output which is generally generated as the result of some action being performed on objects. However a simple assertion describing

a fact may not have an output. The *'output'* may also contain examples or use of science concepts. For example looking again at our sample definition of *'hydroelectric power'* the word that is categorized as *'output'* is *'electricity'*, since it is the final product of the process as suggested in the sentence.

The sequence we are looking for is derived by arranging the words in each of these classes in the following sequence *'Initiator; conditions; action; action-on; output'*. It should be noted here that in this sample definition of *'hydroelectric power'*, the sequence of words arranged in the sentence is different than the logical arrangement of words with respect to the sequence of events or objects.

3.3.2 Classifier

The classification of the words into these classes requires Natural Language Processing (NLP) techniques, because the sentences fetched from the textbook are English sentences and are written as Natural language with no particular single structure. This text is a random mix of different type of sentences, which are used to explain a single science process or concept. Although all the sentences do not follow a single structure but we may assume that all the sentences in a textbook are grammatically correct. Therefore the rules of English grammar are applicable to these sentences and we can exploit these rules for Information Extraction form the text. That is, for classification of words into the above defined five classes which further leads to sequence generation.

In order to use the grammatical structure of the sentences for IE we need a dependency parser which can correctly read and process these sentences. A dependency parser represents dependencies between words and these dependencies are based on the structure of sentence. Out of multiple dependency parsers present we picked Stanford Dependency parser (SDP) as it is more robust and accurate than some of other dependency parsers like Minipar and Link Parser [14]. Also the fact that SDP is currently evolving with new versions and have a very good support for users further convinced us to use it. SDP gives typed-dependencies which can be used for classification of words. Typed-dependency is given for each word pair in a branch of a phrase tree structure this can be utilized in deciding the class for each individual word of a sentence. Before getting into implementation of SDP for classification we should understand the output of SDP and how it works, in much detail.

3.3.3 Stanford Dependency Parser (SDP)

Stanford Dependency Parser provides typed dependencies between a pair of words in a sentence that are related to each other. The type of dependency is extracted from the parse tree of the sentence using rules or patters applied on the phrase structure parse. For the dependency generation each node of parse tree is matched, to the pattern of dependencies and the matching pattern with most specific grammatical relation is assigned as dependency type. About 53 grammatical relations [60], (which are arranged in a hierarchical manner, rooted with most generic relation) are currently present in SDP. For example as given in [14] "the dependent relation can be specialized to *aux* (auxiliary), *arg* (argument), or *mod* (modifier). The *arg* relation is further divided into the *subj* (subject) relation and the *comp* (complement) relation, and so on". Following is the whole hierarchy of these grammatical relations:

```
root - root
dep - dependent
    aux - auxiliary
        auxpass - passive auxiliary
        cop - copula
    arg - argument
        agent - agent
        comp - complement
```

        - acomp - adjectival complement
        - attr - attributive
        - ccomp - clausal complement with internal subject
        - xcomp - clausal complement with external subject
        - complm - complementizer
        - obj - object
            - dobj - direct object
            - iobj - indirect object
            - pobj - object of preposition
        - mark - marker (word introducing an advcl)
        - rel - relative (word introducing a rcmod)
    - subj - subject
        - nsubj - nominal subject
            - nsubjpass - passive nominal subject
        - csubj - clausal subject
            - csubjpass - passive clausal subject
- cc - coordination
- conj - conjunct
- expl - expletive (expletive there)
- mod - modifier
    - abbrev - abbreviation modifier
    - amod - adjectival modifier
    - appos - appositional modifier
    - advcl - adverbial clause modifier
    - purpcl - purpose clause modifier
    - det - determiner
    - predet - predeterminer
    - preconj - preconjunct
    - infmod - infinitival modifier

mwe - multi-word expression modifier
partmod - participial modifier
advmod - adverbial modifier
neg - negation modifier
rcmod - relative clause modifier
quantmod - quantifier modifier
nn - noun compound modifier
npadvmod - noun phrase adverbial modifier
tmod - temporal modifier
num - numeric modifier
number - element of compound number
prep - prepositional modifier
poss - possession modifier
possessive - possessive modifier ('s)
prt - phrasal verb particle
parataxis - parataxis
punct - punctuation
ref - referent
sdep - semantic dependent
xsubj - controlling subject

SDP uses Stanford parser [61] to generate phrase structure of sentence. Stanford parser is a statistical parser that is trained on *Penn Wall Street Journal Treebank*.

While generating dependencies SDP assigns a root node which is the head of the tree. SDP tries to assign a verb as a root as and when possible, but this is not a rule. Further dependencies are generated in the form of a tree, which is a "singly rooted directed acyclic graph with no re-entrances" [14]. A sample dependency graph is shown in Figure 3.2. In order to cater sentences that are questions, SDP is trained with a separate set of questions that are not a part of the *Penn Wall Street Journal*. SDP also provides a collapsed dependency version of the dependency tree, wherein it collapse a pair of dependencies into a single typed dependency, labeled with the name of the dependency collapsed. This is generally done with prepositions and conjunctions [60].

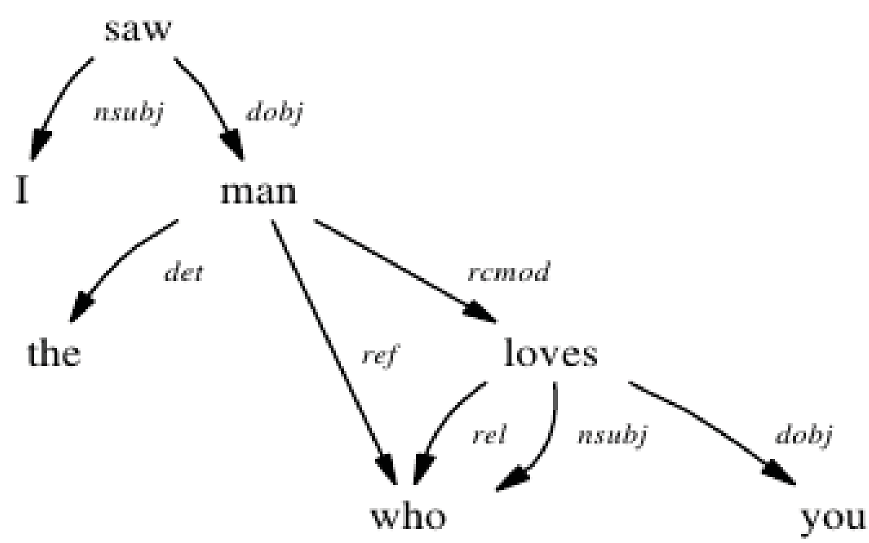

Figure 3.2. A typed dependency parse for “I saw the man who loves you” [14].

3.3.4 Implementation of SDP

SDP is implemented out of the box (without any customizations) in our system, however we are using a collapsed version of the dependency graph representation. The collapsed version adds some extra prepositions to the dependency type list [60], this gives us more information about the type of preposition used. A list of two-word prepositions that SDP can collapse is given in Table 3.1. Since the classifier build over the result of SDP uses only the dependency to decide the class of dependent word, therefore more details in the dependency can help in better classification of dependent word. A dependency representation of SDP is a triplets "*dependency (governor-number1, dependent-number2)*". The dependency here is a short form of the typed-dependency between the governor and the dependent word, the integer representing '*number1*' gives the position of governor words in the sentence and similarly the integer representing '*number2*' gives the position of dependent words in the sentence. SDP when executed on a sentence gives a list of those triplets one for each pair of governor and dependent word. The list of dependencies for our sample sentence definition of '*digestion*' which is "*chemical and mechanical process that breaks food down into small molecules so that they can be absorbed by the body.*" is shown in Table 3.2, while the parse tree used to develop this dependency listing is shown in Figure 3.3.

Table 3.1
Two-word prepositions Stanford Dependency Parser can Collapse [60].

according to	as per	compared to	instead of	preparatory to
across from	as to	compared with	irrespective of	previous to
ahead of	along with	alongside of	apart from	as for
as from	as of	aside from	away from	based on
because of	close by	close to	contrary to	due to
depending on	except for	exclusive of	contrary to	followed by
inside of	next to	near to	off of	out of
outside of	owing to	preliminary to	prior to	pursuant to
regardless of	subsequent to	such as	thanks to	together with

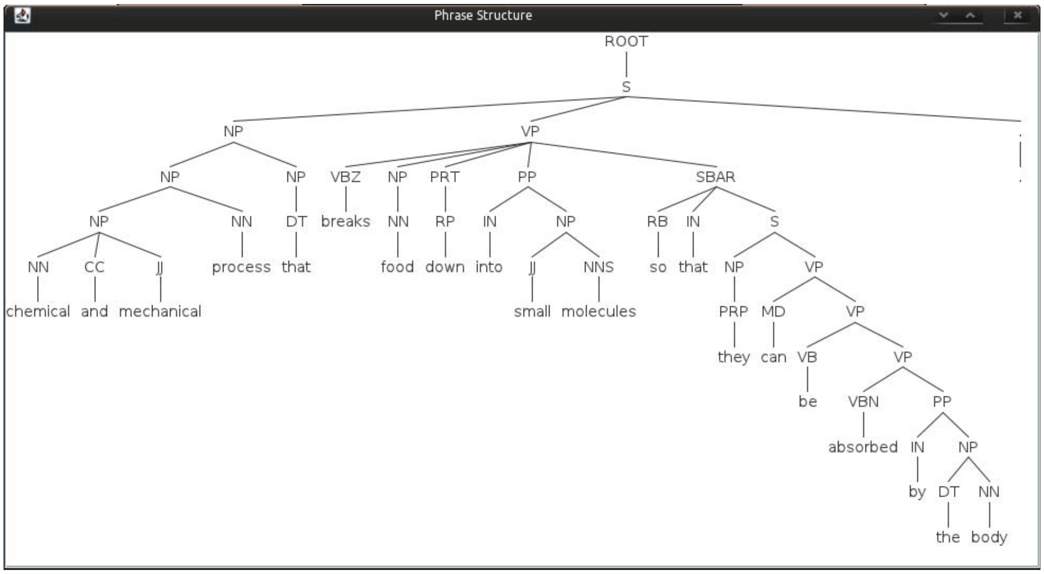

Figure 3.3. Phrase tree structure of a sample sentence: "chemical and mechanical process that breaks food down into small molecules so that they can be absorbed by the body".

In our implementation, SDP is run as a batch on all the sentences that are extracted from the textbook for a single concept as described in section 3.2. SDP works on a single sentence at a time and is very good in identifying sentence boundaries in a text. Therefore all the sentences are provided as input at one time to SDP but on the other hand once SDP generates dependency listing, then output is processed per sentence. This is ensured by the using two consecutive line feeds as delimiter in the dependency list.

The classifier as discussed in the following section also considers POS tag of individual words. We are taking POS tags of each word in the sentence from the output of SDP. POS tags are given by SDP along with typed dependency listing. This is done by using option the "*WordsAndTags*" along with "*typed Dependencies*" while executing SDP. SDP assigns POS tags by considering whole sentence and does a bidirectional matching as in Stanford Tagger [50]. Therefore the result of SDP's POS tagging is

Table 3.2
Dependency list for definition of "*digestion*".

nn(process-4, chemical-1)
conj_and(chemical-1, mechanical-3)
nn(process-4, mechanical-3)
nsubj(breaks-6, process-4)
dep(process-4, that-5)
root(ROOT-0, breaks-6)
dobj(breaks-6, food-7)
prt(breaks-6, down-8)
amod(molecules-11, small-10)
prep_into(breaks-6, molecules-11)
advmod(absorbed-17, so-12)
dep(absorbed-17, that-13)
nsubjpass(absorbed-17, they-14)
aux(absorbed-17, can-15)
auxpass(absorbed-17, be-16)
advcl(breaks-6, absorbed-17)
det(body-20, the-19)
agent(absorbed-17, body-20)

different from conventional taggers. POS tags assigned by SDP are separated from the word by a '/' as shown below for definition of "digestion":

"*chemical/NN and/CC mechanical/JJ process/NN that/DT breaks/VBZ food/NN down/RP into/IN small/JJ molecules/NNS so/RB that/IN they/PRP can/MD be/VB absorbed/VBN by/IN the/DT body/NN.*"

3.3.5 The Classifier

The classifier which finally segregates different words of the sentence into five classes is a rule based classifier which works on the output from SDP. The output from SDP is a list of dependencies which can be viewed as a directed graph having a root, which is the same word, assigned by SDP as *root*. The root word leads to another dependent word, for example in the sample dependencies generated from SDP as shown in Table 3.2 the dependent words on root word ('*breaks*') are '*down*', '*molecules*', '*process*' and '*food*'. This dependency representation can be looked as a directed graph, Figure 3.4 shows this directed graph representation of dependencies shown in Table 3.2. This graph is generated using grammar scope which provides a graphical interface to Stanford Dependency Parser.

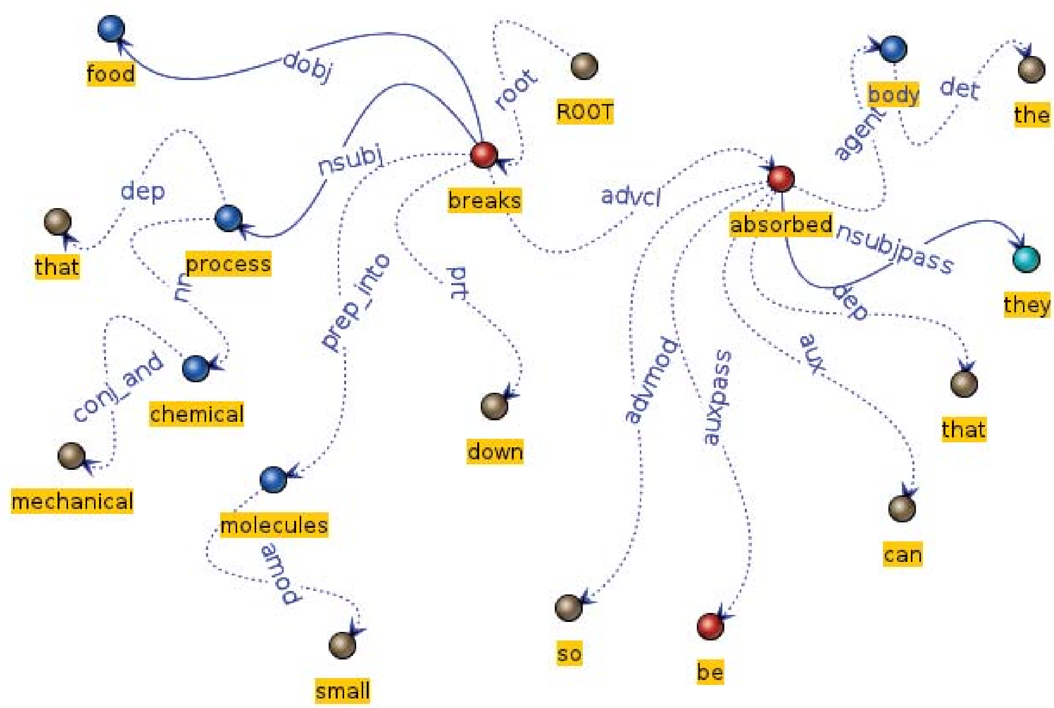

Figure 3.4. Dependency graph generated using grammar scope for sentence "chemical and mechanical process that breaks food down into small molecules so that they can be absorbed by the body".

Let us have a closer look at the navigation and dependency arrangement in the directed graph for our sample definition of '*digestion*' as shown in Figure 3.4. We start from root and then we can proceed to other words which are dependent on the root. Here 'break' is assigned as *root*. In this dependency list the words that are dependent on '*break*' are '*process*' (nsubj(breaks-6, process-4)), '*food*' (dobj(breaks-6, food-7)), '*down*' (prt(breaks-6, down-8)), '*molecules*' (prep_into(breaks-6, molecules-11)) and '*absorbed*' (advcl(breaks-6, absorbed-17)). These four words are directly linked to '*breaks*' in the dependency graph. Moving ahead one step we can reach to other words that are dependent on the above listed words derived from '*break*', for an example words '*that*' and '*chemical*' can be directly reached from the dependencies whose *governor* word is '*process*' (nn(process-4, chemical-1), nn(process-4, mechanical-3), dep(process-4, that-5)). Similarly we can navigate to all the remaining words in the sentence following the *governor* words in dependency listing.

The dependencies are assigned to the pair of words depending on the English grammar, so a word which is dependent on another word can be allocated to a class if the *governor* word is itself already allocated to a class. This allocation is driven by the dependency type, since we know that for a same type of sentence structure always the same dependencies will be allocated. This led to generation of rules which are based on *dependency* type and the class of *governor* word. Each rule is a combination '*class of governor*', '*dependency*', '*class of dependent*', '*Part of Speech of dependent*' (optional). Here '*dependency*' is the typed dependency assigned by SDP, '*class of governor*' is one of the class out of '*Initiator*', '*Condition*', '*Action*', '*Action-on*' and '*Output*' in which the *governor* word is already present, '*class of dependent*' is the class which the dependent word should be allocated to. Some of the rules also take into consideration the part of speech of the *dependent* word, that is, the rule may suggest one class for a *dependent* word if *dependent* word is a verb and may suggest another class if the *dependent* word is a noun. One of the sample rules that have a part of speech in it is 'Action; advcl; Action; verb' Here the first *Action*' suggests the class of *governor* word, '*advcl*' suggests the dependency '*adverbial clause modifier*', second '*Action*' refers to the class in which the dependent word should be placed and '*verb*' suggests that this rule is applicable only if the *dependent* word is a verb. A list of all the rules prepared for this classifier is shown in the Appendix.

The classifier navigates the dependencies starting from root until all the dependent words are classified into their respective class. At each dependency it searches for a relevant rule and allocates a class to *dependent* word. We can see that the class of a word is derived using typed dependencies assigned by SDP. Figure 3.5 shows the automaton representation of the rules. In order to start classifying we start at the root and depending on the part of speech of root word we assign it to either '*action*' class or '*condition*' class (verbs are '*action*' and nouns are '*condition*'). Once we get the class of root word we can proceed further in the graph and classify all the words in a sentence, by picking applicable rules and applying them for allocation of dependent word to a class.

The final output of the classifier is a list of words for each of the five classes. For a single concept the list contains words from all the sentences that were fetched for that science process/concept. This is done by appending the list for each sentence. While appending words from multiple sentences of the same science concept, many words are bound to occur multiple times and some of them may occur multiple times in different classes as well. This is admissible as in later phases the weightage is applied for every word in each of the lists. Table 3.3. Shows the final output of our sample sentence, which is definition of "*digestion*". The classified words shown in Table 3.3 contain multiple stop-words (refer to section 2.2 for the list of English stop-words), which will be removed in later phases. The reason for keeping stop-words in this Phase is that, any of these words could be a link to traverse to other words in the dependency graph. Also the list of words does not follow the sequence of words as in the original sentence, because of tokenization done by SDP and since the sequence of words in a phrase is not important in our implementation we are not recreating it. For example the words in class '*Action*' are '*breaks*' and '*down*', in the list of sentences any of these two can come first, there is no ordering. Therefore it can be '*down*' and then '*breaks*' as well. Figure 3.6 describes all the three major steps in Phase-1 processing for a sample concept of precipitation.

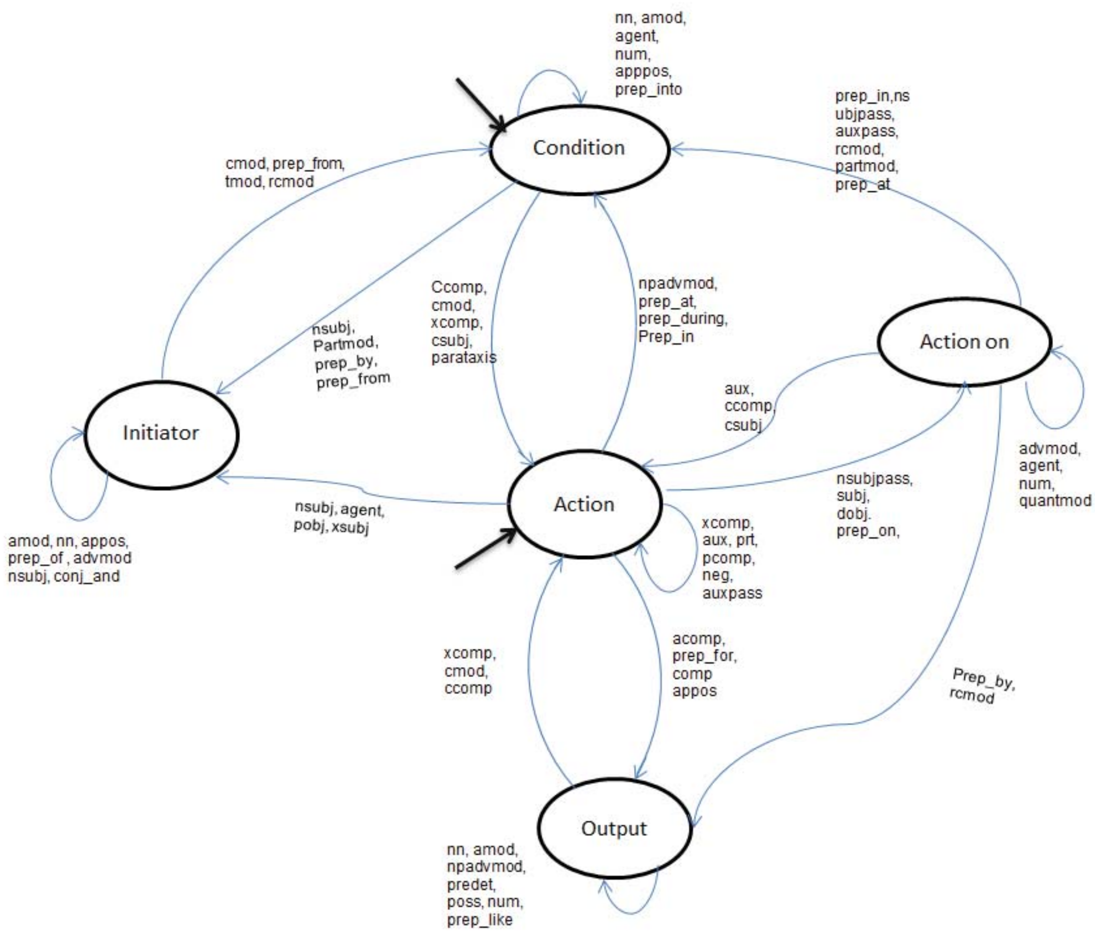

Figure 3.5. Automaton for rule based classifier, with a subset of rules.

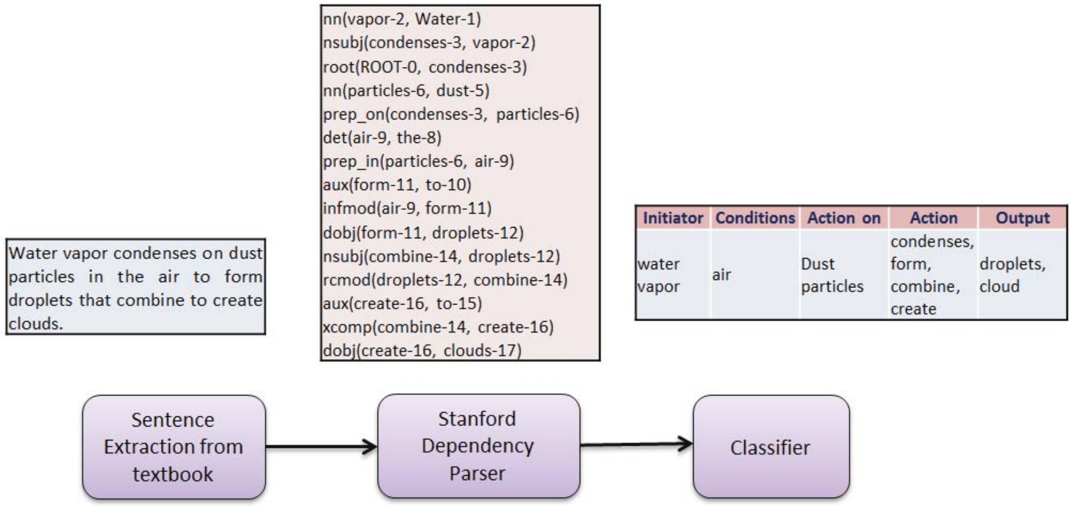

Figure 3.6. Overview of processing of Phase-1 with an example of "*Precipitation*".

Table 3.3
Classification of words in definition of "*digestion*".

Initiator	**Condition**	**Action**	**Action-on**	**Output**
process	absorbed	breaks	food	small
chemical	body	down		molecules
mechanical	so			
they	be			
	can			
	that			
	the			
	that			

3.4 Phase-2: List of Atomic-Sound Generation

The sequence of words found from the classifier in Phase-1 is a major ingredient in preparation of audemes. This creates the framework for development of non-verbal micro-narratives. In order to create an audeme we need to find the suitable options for the sounds that can represent the major words in the lists of words in every class. Note that this list also contains stop-words, so the first and foremost step before further processing is to discard these stop-words. An important observation is that as we used collapsed dependencies for dependency list generation from SDP we already have removed some of the stop-words which have been marked as prepositions and collapsed by SDP. A small set of these collapsed words is already shown in Table 3.1. The fact that not all words that are present in a sentence have a non-verbal sound affect associated with them, so we have to find extra words that better relate to atomic-sounds and still convey the same meaning of the word. In order to find the candidate words that have these sound affect associated with them, we look into rich lexical resources. Two of such resources that have been identified and implemented are 1) Online Thesaurus 2) WordNet synsets. The synonyms fetched from both of these resources are consolidated and then each word of this consolidated list is ranked on weightage. The weight of each word is calculated with respect to its correlation to the concept for which the processing has been done, its correlation to sounds and part of speech. The knowledge-base used for finding the correlation to sound is a hand curated database of sound-words created from Roget's thesaurus, it mostly comprises of onomatopoeic words (words that imitates or suggests the source of the sound that it describes) [62]. The top ten words in the ranked list are flagged as most suitable words for sound representation. Words that correspond to the atomic-sound files in sound-word database are directly allocated the respective atomic-sound, while others are flagged and sent to sound experts for atomic-sound allocation. Figure 3.7 shows the overview of processing in of Phase-2.

3.4.1 Removal of stop-words

As already stated above that in Phase-1 classification stop-words were not removed from list of words, but for further processing we do not require stop-words. Also fetching synonyms for stop-words will result in a lot of irrelevant words, because a single stop-word may occur multiple times in the list and if we fetch its synonyms multiple times we will have many non-related words in each class. In order to remove stop-words we used the stop-word list for English in WordNet, this list is also given in section 2.2. All the words are searched for every science concept and if any one of those is found to be a stop-word it is deleted from the list. The resultant list after stop-word removal for our sample sentence (definition of '*digestion*') is shown in Table 3.4.

Table 3.4
List of words for definition of "*digestion*" after stop-word removal.

Initiator	**Condition**	**Action**	**Action-on**	**Output**
process	absorbed	breaks	food	small
chemical	body			molecules
mechanical				

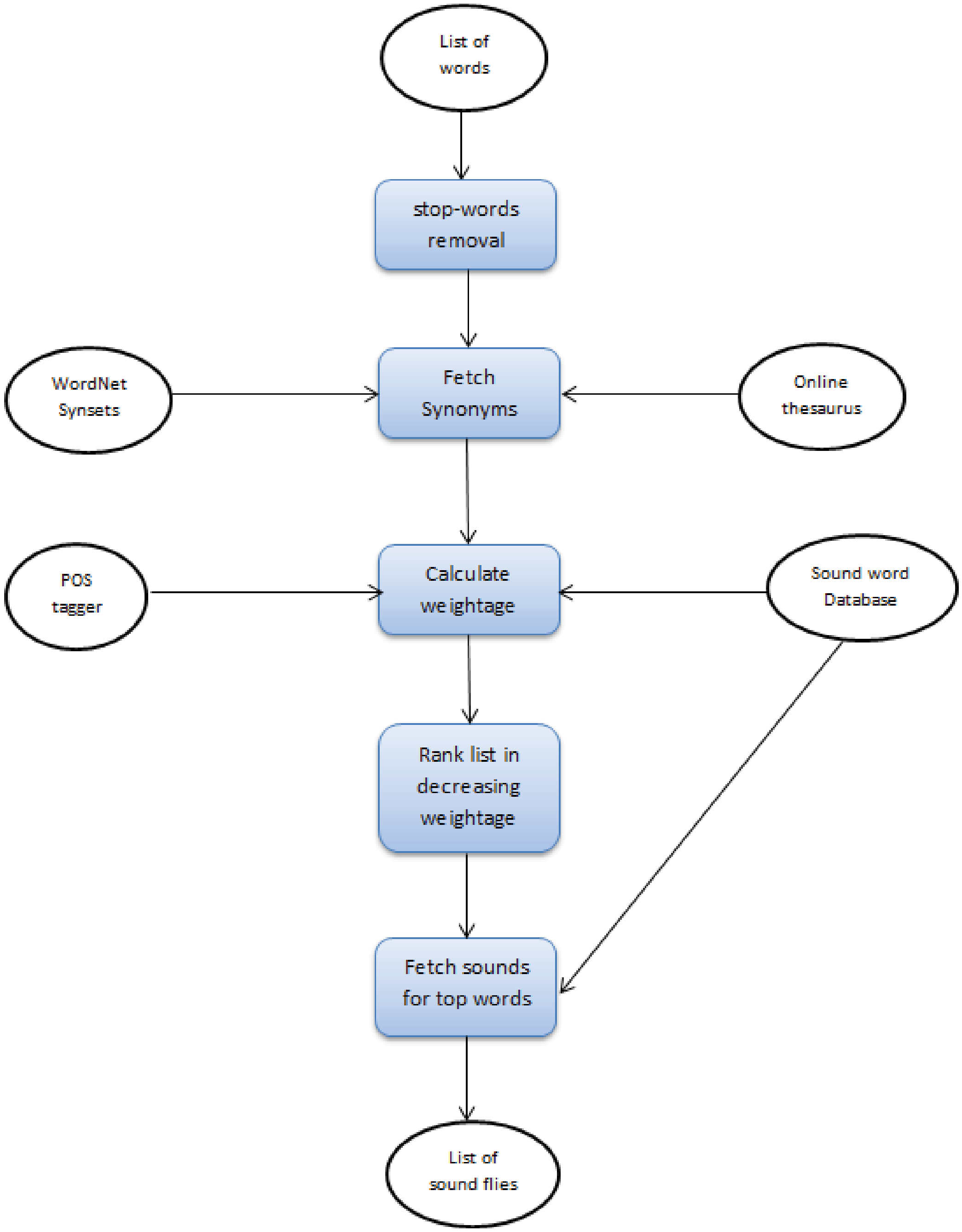

Figure 3.7. Processing in Phase-2.

3.4.2 Synonyms from Online Thesaurus

The words that are used in textbook to describe a process may not be correlated to sounds. In order to get better sound correlated words, a synonym list is generated. This synonym list is generated using two external lexical resources. One of these lexical resources is an online Application Program Interface (API) for thesaurus.com being provided by dictionary.com. This API is accessed through an API key provided by dictionary.com, which is inserted into the URL (Uniform Resource Locator) along with the searched word while requesting the thesaurus result of that word. The online API returns the result in XML format. Figure 3.8 shows the XML output received from thesaurus.com API as viewed in a browser. This XML output is parsed and separate synonym words are identified and extracted from the XML. Since result returned by the API contains definitions and other details as well (as shown in Figure 3.8), we have to make sure that only synonyms are extracted from it. We have used regular expressions and basic level XML parsing to ensure only synonyms are extracted correctly from the XML result. The list of synonyms generated from thesaurus.com is appended to the existing list per class. The synonym words extracted by thesaurus API of dictionary.com for the word "*process*" are shown in Table 3.5.


```
<thesaurus query="process" totalresults="3" page="1">
 - <entry type="direct">
     <headword>process</headword>
     <partofspeech>noun</partofspeech>
     <definition>method; series of actions to achieve result</definition>
   - <synonyms>
       <a href="/browse/action" class="noline">action</a>, <a href="/browse/advance" class="noline">advance</a>, <a href="/browse/case" class="noline">case</a>, ch
       class="noline">course</a>, course of action, <a href="/browse/development" class="noline">development</a>, <a href="/browse/evolution" class="noline">evolu
       class="noline">fashion</a>, <a href="/browse/formation" class="noline">formation</a>, <a href="/browse/growth" class="noline">growth</a>, <a href="/browse
       href="/browse/means" class="noline">means</a>, <a href="/browse/measure" class="noline">measure</a>, <a href="/browse/mechanism" class="noline">mech
       class="noline">mode</a>, <a href="/browse/modus%20operandi" class="noline">modus operandi</a>, <a href="/browse/movement" class="noline">movement<
       class="noline">operation</a>, outgrowth, <a href="/browse/performance" class="noline">performance</a>, <a href="/browse/practice" class="noline">practice
       class="noline">procedure</a>, <a href="/browse/proceeding" class="noline">proceeding</a>, <a href="/browse/progress" class="noline">progress</a>, <a href=
       class="noline">progression</a>, red tape, <a href="/browse/routine" class="noline">routine</a>, <a href="/browse/rule" class="noline">rule</a>, <a href="/brow
       href="/browse/step" class="noline">step</a>, <a href="/browse/suit" class="noline">suit</a>, <a href="/browse/system" class="noline">system</a>, <a href="/bro
       href="/browse/transaction" class="noline">transaction</a>, <a href="/browse/trial" class="noline">trial</a>, unfolding, <a href="/browse/way" class="noline">w
       class="noline">wise</a>, <a href="/browse/working" class="noline">working</a>
     </synonyms>
   </entry>
```


Figure 3.8. XML output generated by thesaurus.com API.

Table 3.5
Synonyms from online thesaurus for the word "*process*".

action	advance	case	course	development	prepare
evolution	fashion	fashion	formation	growth	refine
manner	means	measure	mechanism	mode	transform
movement	operation	performance	practice	procedure	treat
proceeding	progress	progression	routine	rule	activity
stage	step	suit	system	technique	agility
transaction	trial	way	wise	working	alacrity
alter	concoct	convert	fulfill	handle	alertness
animation	ballgame	business	bustle	commotion	dash
deal	energy	enterprise	flurry	force	functioning
game	going	happening	haste	hopper	industry
life	liveliness	motion	movement	occupation	operation
plan	power	process	proposition	racket	reaction
response	rush	scene	spirit	stir	stunt
trip	turmoil	vigor	vim	vitality	vivacity

3.4.3 Synonyms from WordNet

As synonyms are fetched from online thesaurus, likewise we are also extracting synonyms from WordNet synsets as well. Synsets are sets of synonyms used by WordNet to represent a word sense, and synonym is a symmetric relation between word forms [15]. All the words that are present in the synset containing the word (each word after Phase-1) are extracted and added to the list already created in section 3.4.2. We have implemented both "wordnet" and "verbnet" presnt "nltk.corpus" module of NLTK. WordNet synsets also contain information about *Hyponyms* (sub-name), *Hypernyms* (super-name), *Meronyms* (part-name), *Holonyms* (whole-name), *Troponym* (manner-name for verbs). These extra words can also be extracted in the same way as synonyms, since they all are somehow related to the word being processed. But for our current implementation and for the sake of only fetching synonyms these words have been discarded while synonym list generation. The words fetched from WordNet synsets for the sample word "*process*" are shown in Table 3.6.

Table 3.6
Synonyms extracted from WordNet for the word "*process*".

physical_process	sue	mental_process	march	procedure
swear_out	work	cognitive_process	work_on	outgrowth
unconscious_process	appendage	serve	operation	action
cognitive_operation	litigate	process	summons	treat

3.4.4 Sound-word database

Before going further into the final selection of sounds from the synonyms we should fist explain the sound-word database. Since it is used to assign weight as well as to select sounds for a word. The sound-word database is the database that contains the atomic-sounds with a list of tags associated with those sounds. These atomic-sounds are non-speech sounds taken from daily experiences and have the potential to represent a set of related actions, things or modifiers. These atomic-sounds when together arranged in a sequence form an audeme. Atomic-sounds are created by sound experts and inserted into the sound-word database. The actions, things and modifiers associated with an atomic-sound are assigned as tags to each of these atomic-sounds. For example the sound of '*thunder*' can represent '*rain*', '*cloud*', '*rage*', '*deluge*, '*storm*', '*power*', '*torrent*', '*lightning*' etc. The sound-word database contains a list of atomic-sounds, each of them has multiple tags associated with them. These tags form the sound-words or in other words, each sound-word is known to have a sound representation associated with it which is actually the atomic-sound itself. This database including both sounds and tags is hand curated, by sound experts. The tags present in this database are stemmed words, this has been done for effective matching purposes. This is an incremental database and suggestions for the sound-words are given by the system to sound experts. The suggested words are generated as explained in following section.

3.4.5 Weightage and Ranking

After fetching synonyms for all the words that were present in each of five classes after section 3.4.1 that is after stop-word removal, they are ranked by weightage of words in that list. This weight is primarily based on the relative frequency of occurrence of a word in a single class. Relative frequency of occurrence is the number of times a word occurs in a class divided by the total number of words in that class. While calculating the frequency of a word inflections of a word are taken into consideration.

$$RelativeFrequency = \frac{number\ of\ occurrences\ of\ a\ synonym\ in\ a\ class}{total\ number\ of\ synonyms\ in\ a\ class} \tag{3.1}$$

Before assigning the ranking of words on the basis of their initial relative frequency of occurrence, additional weights are assigned to each word based on couple of other criteria as well. These criteria are:

1. A word's correlation to the sounds present in sound-word database.
2. A word's Part-of-Speech.

Sound-word correlation: The correlation of every word with a known atomic-sound is taken into consideration by searching it in the tags field of sound-word database.

As explained in section 3.4.4, tags in sound-word database contain the words that can represent that atomic-sound. This atomic-sound can be directly picked up to represent the word if the word is present as a tag in sound-word database. Before searching the word in sound-word database stemming is applied to each word. This is done to ensure that there is no mismatch of words only because of inflections, since inflections rarely change the associated sound with a word. Since we are using Natural Language Toolkit for most of our processing and Porter's Stemmer is available in NL Toolkit (Lancaster Stemmer is also present in NLTK), so we are using the version of Porter's Stemmer provided in NLTK [46]. If a word is found as a tag in sound-word database then that word is supposed to get a higher weight in the list of words. This will ensure that the words that have a higher correlation to an atomic-sound will be preferred for sound representation. Higher weightage is ensured by increasing the frequency of occurrence of a word, since final ranking criteria is frequency. So any word that has a hit in the sound-word database is appended once more in the list of words, thus suggesting higher weight.

Part-of-Speech: The second criterion for weightage is part-of-speech of a word. This is done because of the observation as already discussed in section 3.3.1. that is, actions tend to give sound to objects. So in order to find the correlated sound for an object it is better to look for the actions that are associated with that object. A simple example of '*cup*' is already given in section 3.3.1. To ensure that actions are given a higher weight while ranking words for sound representation, we are checking the part-of-speech (POS) of every word in the synonym list and if POS is found to be a verb or verb related then its frequency is further increased in the list.

The ranked list of words is obtained by ranking them on the final weightage obtained by applying the above described conditions. Thereafter, top 10 words per class are considered to portray a sound. It should be noted that not all these top 10 words would have a sound associated with them in the current state of sound-word database, but still they are most probable words for representation of an action, thing or modifier under consideration for a concept. These top ten words are stored for further atomic-sound selection.

Each of the ten words selected from the ranked list above is matched to the words that are present as tags in sound-word dictionary. This is again done after stemming the words. If the stemmed word is present as a tag in sound-word database then corresponding atomic-sound is picked for that word. If not, then that word is flagged as suggestion for creation of new atomic-sound to sound experts. Thus ensuring that sound-word database is incremental with respect to coverage of concepts and amount of sounds per concept. These sounds are then arranged in individual classes as shown in Tables 3.7 and 3.8 for a single word per class. It is worth noting that a single tag word can be associated with multiple sounds.

Table 3.7
Phase-2 output for "*digestion*".

Initiator	**Condition**	**Action**	**Action-on**	**Output**
Mechanical_car	Absorb_suck	Break_crack	Food_chew	small_crumb
Mechanical_cycle	Absorb_drink	Break_split	Food_gulp	small_baby
Mechanical_grind	Absorb_soak	Break_tear	Food_cook	small_break
Mechanical_hammer		Break_clash	Food_drink	
			Break_occasion	

Table 3.8
Phase-2 output for "*precipitation*".

Initiator	**Condition**	**Action**	**Action-on**	**Output**
steam_engine	air_breath	combine_band	dust_flakes	cloud_haze
steam_force	air_wind	combine_fuse	dust_ashes	cloud_thunder
steam_fizz	air_whiff	combine_heap	dust_soot	cloud_puff
steam_bubble	air_breeze	combine_bond	dust_earth	cloud_steam

3.5 Phase-3: Audeme Generation

The list of sounds that is obtained after Phase-2 of processing contains the list of atomic-sounds that can be directly used to create an audeme. But these atomic-sounds are in a large number per class. This is due to multiple factors, one of the obvious factor is that there are multiple words in a class (ten here), so there has to be multiple sounds (at least one per word). But as shown in Table 3.7 and 3.8, a single word can be present as a tag in multiple atomic-sounds; therefore the total number of atomic-sounds being picked up is more than ten. To understand the reason for this, let us take the example of word '*steam*' which is a synonym of '*water vapor*' which was present in the definition of 'precipitation'. As shown in '*initiator'* of Table 3.8. There can be multiple sounds associated with '*steam*' which is due to multiple uses and conditions in which '*steam*' is known to be present. For example steam can be associated with the sound of '*steam Engine*' or '*bubbles*' or '*sound of sudden release of gas*' all suggesting '*steam*'.

As already seen that multiple atomic-sound files are fetched from sound-word database as probable atomic-sounds in a class, but in these atomic-sound files there are chances of duplication as well, which is due to the fact that out of the ten words per class more than one word may be present as a tag for a single atomic-sound. For example '*combine*', '*create*', '*form*' are '*action*' words for '*precipitation*' but they all are tagged in atomic-sound for '*fuse together*'. The obvious solution for this problem is to rank atomic-sounds within a class after all the sounds of top 10 synonym words are selected for a concept. This rank is stored as a number which is unique for every combination of concept and atomic-sound, this rank is then used as a factor for selection of atomic-sound per class in an audeme. This ranking is initially done on the basis of the count of occurrences of an atomic-sound per class for a single science concept.

3.5.1 Correlation Factor

Audemes for a concept can be created by selecting a single sound from each class out of multiple atomic-sounds selected above and then playing them in the sequence: "initiator;condition;action;action on;output". The task of selection of the best sound that can represent each class is crucial in the creation of correct audeme. This is done by using correlation factor which is an integer associated with every atomic-sound in a class for a science concept. Atomic-sound with highest correlation factor is selected per class for audeme representation. The initial value of correlation factor is the count of occurrences of an atomic-sound per class as already stated.

By conducting a series of experiments at Indiana School of Blind and Visually Impaired (ISBVI) it has been found that visually impaired students in general are better to identify individual sounds in a combination. So the audeme that we perceive to be good for a science concept/process may not be the best for them. Moreover the system explained up until now do not ensure the best selection of atomic-sound. Even for the students with sight the perceptions of sound may vary depending on cultures and personal experiences. Therefore, the selection of sound on the basis of the count of occurrences of atomic-sound may not give the best audeme.

In order to make the audemes generalized and more suited to the preference of actual users, which in this case are BVI students, we propose a feedback polling mechanism. This mechanism is used for the dynamically ranking atomic-sounds per class per concept. This is designed to be based on a correlation factor. As more and more feedback or data are collected from the users who interact with audemes, either in the form of games or quizzes, the correlation factor is updated. This updated correlation factor is then used for selection of relatively better atomic-sound per class. The atomic-sound with highest correlation factor is picked up as the best atomic-sound to represent a concept in a class. An example of ranked list of atomic-sounds based on correlation factor is shown in Table 3.9.

Table 3.9
Atomic-sounds with correlation factor (integer separated by ':').

Initiator	**Condition**	**Action**	**Action-on**	**Output**
A1:3	B1:5	C1:4	D1:2	E1:5
A2:2	B2:3	C2:3	D2:2	E2:2
A3:1	B3:2	C3:3	D3:1	E3:2
A4:1	B4:1	C4:2	D4:1	E4:1

3.5.2 Updating Correlation Factor

The audemes generated using the initial values of correlation factor are selected for further uses. Students at blind schools play and interact with these audemes via games based on touch interface on handheld devices or by the help of instructors in classrooms. We record the choices made by players, to select the best audeme out of the given options. This information is the total count of the number of times an audeme was perceived to be correctly representing a science process/concept in a game play by multiple players. In order to actually present users with multiple options of atomic-sounds for a concept we randomly select atomic-sounds from the top ten already ranked atomic-sounds (this ranking is based on the correlation factor). The count that has been recorded per audeme for a concept is further used to calculate and change the correlation factor of an atomic-sound for a concept. The change in correlation factor changes the ranking of atomic-sounds in a class, therefore next time more relevant atomic-sound is presented by the in the audemes for a concept. The calculation of new correlation factor is done by adding individual counts per atomic-sounds from all the audemes. To understand this let us consider four audemes made out of atomic-sounds as shown below.

Audeme #1: A2 + B1 + C3 + D2 + E4
Audeme #2: A3 + B2 + C3 + D1 + E2
Audeme #3: A4 + B4 + C1 + D2 + E3
Audeme #4: A2 + B3 + C2 + D3 + E4

Consider that 30 players played a puzzle game to identify the best audeme for a science concept. This game has above mentioned four options of audemes and one of them has to be selected as the correct audeme (we can always have an audeme containing highest ranked atomic-sounds from each class, which is perceived by the system to be correct audeme). Suppose the total count of the number of students choosing each audeme as correct option is as shown below in parenthesis, let us call it as *count*:

Audeme #1: A2 + B1 + C3 + D2 + E4 (12)
Audeme #2: A3 + B2 + C3 + D1 + E2 (8)
Audeme #3: A4 + B4 + C1 + D2 + E3 (4)
Audeme #4: A2 + B3 + C2 + D3 + E4 (6)

The number of times an audeme is perceived to be correct by players is used to increase the correlation factor of individual atomic-sounds. This is done by adding the *count* for each atomic-sound present in each audeme. For example as shown below the increase in the correlation factor of atomic-sound 'A2' is calculated by adding count of Audeme #1 and Audeme #4; since A2 is present in these two audemes.

A2: (12 +6), A3 : (8)
B1: (12), B2 : (8), B4 : (4), B3 : (6)
C1: (4), C2 : (6), C3 : (12 + 8)
D1: (8), D2 : (12 + 4), D3 : (6)
E2: (8), E3 : (4), E4 : (12 + 6)

This information when updated in the system will modify the correlation factor and the ranking of audemes in each class for a concept, this updated correlation factor and ranking is shown in Table 3.10.

Table 3.10
Atomic-sounds with updated correlation factors and ranking.

Initiator	**Condition**	**Action**	**Action-on**	**Output**
A2:20	B1:17	C3:23	D2:18	E4:19
A3:9	B2:11	C2:9	D1:12	E2:10
A1:3	B3:8	C1:8	D3:7	E1:5
A4:1	B4:4	C4:2	D4:1	E3:6

The updated correlation factor of individual atomic-sounds from the user's feedback to a concept ensures the best audeme is formed as per the user's preference. In the example taken the correct audeme would be an audeme formed by the combination of A2 + B1 + C3 + D2 + E4 atomic-sounds. The method of using feedback or we can say polling from the responses of users and will ensure that after some time the best possible audemes are produced by the system.

This method relies on that fact that initially the players know the details of the science process/concept whose audeme they are playing with. This is in line with the overall objective of automatic audeme generation, which is to assist visually impaired students in science education; therefore instructors can explain science processes and allow them to hear the audemes, while teaching them in a classroom.

After multiple interactions with teachers at ISBVI it has been found that every instructor has their own methods and they would prefer creating their own custom audemes which better aligns with teacher's explanations. The use of correlation factor to change atomic-sound selection provides this much needed flexibility. An instructor can locally change the correlation factor of a single atomic-sound which he wants to put in the audeme, just by manually updating the correlation factor.

We have seen that the methodologies used for generation of audemes in the system, are flexible as per the requirements. Also the use of both online (thesaurus.com API) and offline (WordNet) lexical resources ensures the robustness of the system with respect to synonym generation. All other methods used are essentially offline processing and thus can be interpolated. In order to avoid reprocessing for the same concepts, the implementation of the methods is done with close database integration that is all the intermediate results for different phases of processing are stored in an instance of MySQL database. This adds to the robustness of the system and saves reprocessing every time a query is made for an audeme of a concept.

4 RESULTS

The methods described for generation of audemes per science process/concept ensures that the quality of audemes generated for a concept will eventually improve, as more and more feedback from actual users is received by the system. The relevance of an audemes to a concept is however very subtle, and depends on an individual's perception. This is because the perception of sound cound differ from one person to another due to varied Intelligence Quotient (IQ) or different cultural effects.

Another fact about audemes is that multiple audemes can correctly portray the same science concept. This is logical as we know that there can be multiple definitions or sentences describing a single concept. Two sentences describing the same concept may differ in their details of the underlying events in a process. For example let us consider two sentences that represent '*solar energy*' as described in Glencoe Blue book: 1) "*Energy from sun that is clean, inexhaustible, and can be transformed into electricity by solar cells.*" 2) "*When you sit in sun, walk into the wind, or sail against an ocean current you are experiencing the power of solar energy.*" Both the sentences although refers to the same concept, but represents two totally different definitions of it. Separate audemes derived from these sentences are bound to vary a lot as a result of different aspects of 'ŝolar energy' described in them. Thus we see that by definition there may not be an absolute best audeme for a science concept.

The following sections discuss the results obtained as in various subsections of methods used to generate audemes in this system:

4.1 Phase-1

4.1.1 Manual Classification

The rules that are used for classification in the rule based classifier as explained in section 3.3.5, are extracted by statistical analysis of manually classified data. Manual classification was done as there was no existing knowledge base for such a semantic classification of sentences used in middle school science textbooks. This was done by middle school science experts. Words present in sentences that are used to describe a science concept in the textbook were placed in one of the five classes based on their semantic value for the concept. This was classification was found to be fairly intuitive for simple sentences, provided one is thorough with the concept at hand. An example of a simple sentence being classified is shown in Table 4.1. On the other hand many complex sentences were found to be hard to classify. One such example of a complex sentence is as shown in Table 4.2 "*But space contains massive clouds of gases, dust, and other debris called nebulae that block part of the starlight traveling to Earth making it more difficult for astronomers to observe deep space*". In this sentence, for describing 'nebulae', the author has combined three features, 1)"massive clouds of gasses, dust, and other debris" 2)"block part of starlight travelling" 3) "making it more difficult for astronomers to observe deep space". However one can argue that feature (3) is an effect of feature (2) of nebulae rather than a direct action of it. Thus the semantic action of (3) is not well defined. This manual classification depends on an individual's understanding and perception of a science concept, therefore the combination of classification from multiple sentences disambiguates this situation caused by complex sentences.

As seen in Table 4.1 and 4.2 manual classification is accompanied by allocation of dependencies for that particular word (shown in column 3). Based on this classification of words, the typed dependencies were mapped as a transition from one class

Table 4.1
Manual classification of a simple sentence describing '*Nebula*': "*Nebula is a large cloud of dust and gas that can break apart into smaller pieces and form stars*".

Initiator	Nebula	nsubj(cloud-5, Nebula-1)
Condition	large, cloud, dust, gas	amod(cloud-5, large-4), root(ROOT-0, cloud-5), nsubj(break-12, cloud-5), prep_of(cloud-5, dust-7), prep_of(cloud-5, gas-9), conj_and(dust-7, gas-9)
Action on	smaller, pieces	amod(pieces-16, smaller-15), prep_into(break-12, pieces-16)
Action	break, apart, from	rcmod(cloud-5, break-12), advmod(break-12, apart-13), nn(stars-19, form-18)
Output	Starts	prep_into(break-12, stars-19), conj_and(pieces-16, stars-19)

to another. A list of transitions was prepared, these transitions were based on the class of governor word and typed dependency in the same way as rules are used for classifying the words. This list is a cumulative list of all the sentences and concepts. This list of transitions was then statistically analyzed for all the transitions used for classification of words. Table 4.3 shows a sample transition list.

A total of 700 sentences for 200 different concepts were manually classified and this classification generated 4495 assertions (as shown in Table 4.3). These assertions were matched and compared to the rules generated for classifier. These rules matched to 71.87 percent of the assertions, thus giving the precision of applicability of rules.

Table 4.2
Manual classification of a complex sentence describing '*Nebula*': "*But space contains massive clouds of gases, dust, and other debris called nebulae that block part of the starlight traveling to Earth making it more difficult for astronomers to observe deep space*".

Initiator	space, massive clouds, gases, dust, other debris	nsubj(contains-3, space-2), amod(clouds-5, massive-4), nsubj(called-14, clouds-5), prep_of(clouds-5, gases-7), prep_of(clouds-5, dust-9), conj_and(gases-7, dust-9), amod(debris-13, other-12), prep_of(clouds-5, debris-13), conj_and(gases-7, debris-13)
Condition	contains	root(ROOT-0, contains-3)
Action on	part of starlight, Earth	nsubj(observe-32, part-18), prep_of(part-18, starlight-21), prep_to(traveling-22, Earth-24)
Action	block, traveling, making, called	nn(part-18, block-17), partmod(starlight-21, traveling-22), xcomp(traveling-22, making-25), ccomp(contains-3, called-14)
Output	more difficult for astronomers to observe deep space, nebulae	advmod(difficult-28, more-27), xcomp(making-25, difficult-28), prep_for(difficult-28, astronomers-30), aux(observe-32, to-31), ccomp(called-14, observe-32), xcomp(called-14, observe-32), amod(space-34, deep-33), dobj(observe-32, space-34), acomp(called-14, nebulae-15)

4.1.2 Classifier performance

In order to measure the performance of rule based classifier we compared the words in classes as classified by the rule based classifier and those generated from manual classification process, which is also the gold standard data. As shown in Table 4.4. for

Table 4.3
A subset of the transitions extracted from manual classification, these transitions match to every single word being classified.

Dependency	**class of governor**	**Class of dependent**
amod	Initiator	Initiator
nsubj	Conditions	Initiator
quantmod	Conditions	Conditions
dep	Conditions	Conditions
number	Conditions	Conditions
appos	Initiator	Conditions
prep_in	Conditions	Conditions
nsubj	Action	Conditions
amod	Conditions	Conditions
prep_as	Action	Conditions
nsubj	Action	Conditions
nsubjpass	Action	Conditions
num	Conditions	Conditions
dobj	Action	Conditions

a sentence used to describe “lymphatic cells”: “*The lymphatic system helps protect you from infections and diseases.*” The overall precision with respect to individual words in a class was found to be low due to complex sentences (which contain more words than simple sentences).

The fact that activity of classification of words into five classes is done, only to obtain the sequence of words and therefore the position of a word in a class is not of utmost importance. Rather it is the relative position of a word in the final sequence with respect to other words is what matters. Since, finally the audemes are sequences of

Table 4.4
Comparison of manual and rules based classification for a sentence describing “lymphatic cells”: “*The lymphatic system helps protect you from infections and diseases*”.

Class	**Manual Classification**	**Rule based**
Initiator	lymphatic system	System, lymphatic
Conditions	from infections, diseases	infections, diseases
Action on	You	You
Action	helps protect	helps, protect
Output		

atomic-sounds, therefore this sequence only matter for the quality of audemes. So the performance of the classifier is rather measured by comparing the sequence of words being generated by the classifier with that being formed by manual classification. This comparison resulted in a precision of 75.75 percent. An example of a sentence where exact word classification by rule based classifier is different with respect to the gold standard, but the sequence being generated is exactly same as that of the gold standard classification is shown in Table 4.5. In this example since all the words classified in gold standard to the class “action on” are rather allocated to “output” class in rule based classifier. But the final sequence of words is same in both classifications, when arranged in the order of “initiator;condition;action;action on;output”.

A notable observation to be made in Table 4.5 is, that there is no guarantee that all the five classes are populated with words for a single concept. In other words, sentences that define a process or generally one which represents a fact or assertion may not have words in some of the classes. For example in the sentence classified as shown in Table 4.5 there are no words in class “action_on” as classified by rule based classifier. But since the final classification of sentences for a concept contains

Table 4.5
Sequence comparison between manual classification and rules based classification for the sample sentence "*Muscles and ligaments attach to some of the bumps and pits*".

Class	**Manual Classification**	**Rule based**
Initiator	Muscles, ligaments	Muscles, ligaments
Conditions	from infections, diseases	infections, diseases
Action on	some, bumps, pits	You
Action	Attach	attach
Output		some, bumps, pits

consolidated words from multiple sentences used for the science concept this generally ensures the presence of words in each of the classes.

An example suggesting the reason for this condition is the dependency 'dobj(direct object)' which gives direct object of a verb. This object can be an Action-on or an Output depending on the verb itself. For example one of the sentences used to describe 'precipitation' in Glencoe Blue Book is "*Water vapor condenses on dust particles in the air to form droplets that combine to create clouds.*" The dependencies generated for this sentence are shown in Table 4.6. We know from our understanding of precipitation that an "output" word in this sentence is 'cloud' which is assigned as dobj of 'create' (the 'action') by SDP, but dobj of 'form' (also 'action') is droplet' which is rather an 'action on' word.

The performance of rule based classifier depends heavily on the capability of Stanford Dependency Parser to correctly identify the dependency between two word pair. Some of the typed-dependencies like 'dep (dependent)' are an unknown dependencies, so it is hard to specify the class of dependent word in these cases. Another known limita-

Table 4.6
Dependency list for a sample sentence describing "*precipitation*": "Water vapor condenses on dust particles in the air to form droplets that combine to create clouds".

nn(condenses-3, Water-1)
nn(condenses-3, vapor-2)
nsubj(particles-6, condenses-3)
prep(condenses-3, on-4)
pobj(on-4, dust-5)
root(ROOT-0, particles-6)
prep(particles-6, in-7)
det(air-9, the-8)
pobj(in-7, air-9)
aux(form-11, to-10)
infmod(air-9, form-11)
dobj(form-11, droplets-12)
nsubj(combine-14, that-13)
rcmod(droplets-12, combine-14)
aux(create-16, to-15)
xcomp(combine-14, create-16)
dobj(create-16, clouds-17)

tion of SDP is its incapability to identify and relate words that are comma separated but are conjugate of each other. SDP still performs better than other dependency parsers. Our rule based classifier, tend to minimize the ambiguity in the results of SDP by generating the sequence of words after considering multiple dependencies.

4.2 Phase-2

The result of Phase-2 of processing which is the generation of ranked list of atomic-sounds by fetching synonyms from external lexical resources have been analyzed manually and are found to be suggestive of the undermining science concept. Below is step by step description of results obtained at various sub phases.

4.2.1 Fetching Synonym

The two external lexical resources being used for fetching related words for the science concept at hand are online thesaurus.com API and WordNet as already described in section 3.4. The relatedness of a word to an atomic-sound in the synonym list (which also contains the base words fetched in Phase-1) is being calculated by its frequency of occurrence, its presence in a list already collected sound words (fetched directly from the sound-word database) and its part-of-speech (action words are more correlated to sound). These three conditions together alter the arrangement of words in the list of all synonyms being as a result of new words from above stated external lexical resources. Let us analyze the results of synonyms and their ranking as obtained for "*initiator*" class of "fossil fuels".

Initiator list without synonyms = resources, nonrenewable, energy, fuels, Fossil, fuels, Fossil, fuels, Fossil, fuel, Coal, abundant, fossil.

Top fifty synonyms for Initiator after sorting on weight with their relative frequency:

fire, means, gas, antiquated, fuel, dated, electricity, fossil, sustain, antediluvian, disused, outmoded, give, provide, food, feed, aged, archaic, find, superseded, supply, early, discarded, antique, old, nourish, relic, superannuated, force, power, energy, juice, support, stock, service, stuff, impression, abundant, charge, foster, dispense, fan, anachronistic, dinosaur, outworn, obsolete, nurse, satisfy, gorge, bygone.

We have shown here only top fifty words in the ranked list of synonyms but this list in total have 437 distinct words. A total of 820 synonym words for "*initiator*" were fetched in this example. Out of this 649 words were present in the synonym list (which also contained words from Phase-1) after fetching synonyms from thesaurus.com.

The final result as obtained from Phase-2 is the list of ranked atomic-sounds for each of the word which is present in top ten of the above shown sorted synonym list. That is: *fire, means, gas, antiquated, fuel, dated, electricity, fossil, sustain, antediluvian.* The sounds fetched from sound-word database for these words are:

ind_fire, A_Fierce, ind_air, ind_animal, A_fun, A_people, A_event, ind_fire, A_electricity.

Note that this list contains repeated sounds which will be initial criteria for calculating correlation factor as explained in section 3.5. After the initial calculation of correlation factor the list of sounds for "*initiator*" of "fossil fuels" is as follows:

ind_fire:2, ind_animal:1, A_fun:1, A_event:1, A_electricity:1, A_Fierce:1, ind_air:1, A_people:1

These sound files are picked up as directly correlated atomic-sounds for representation of the top ten words, which is done by matching the stem of a word with the tags for each word. Table 4.7 shows the entries for these atomic-sounds as present in the current state of sound-word database:

Table 4.7
Snippet of sound-word database for atomic-sounds selection.

atomic-sound	**tags**
A_electricity	electr, energi, develop, industri,
A_event	event, occurr, date, happen,
A_Fierce	fierc, feroci, mean, roar, lion,
A_fun	fun, parti, gather, music, children, play, giggl, laugh, entertain, enjoy, delight, happi
A_people	peopl, social, environ, cultur, gather, public,
ind_air	wind, air, ga, atmospher,
ind_animal	anim, creatur, natur, organ, live, biolog,
ind_fire	energi, heat, hot, fire, electr, power, temperatur,

Currently the sound-word database is in its initial phase and so has comparatively very few entries in it. The total count of atomic-sounds currently present in the sound-word dictionary is 76. This is rapidly growing as more and more words are flagged for sounds as a result of processing of Phase-2. The words that are flagged for sound association for "fossil-fuels" are:

"antiquated, fuel, fossil, sustain, antediluvian, fuel, remains, survive, make, constitute, lump, age, practice, auto, exercise, many, apply, use, car, employ, utilize".

Another aspect of the results of Phase-2 is that, even if there is a lower correlation with the results of synonym list generation, it would lead to more of metaphoric audeme generation. In other words we can say that if two synonyms are semantically farther from each other then there is a higher potential of generating metaphorical atomic-sound. Although the methods presented in this thesis are aimed at micro-narrative generation, but experiments have proved that metaphoric audemes are more effective in retention of a concept as they impose thinking on listeners in order to understand and make interpretations out of audemes [17].

4.3 Phase-3

In Phase-3 the actual audeme is generated from the list of atomic-sounds selected at the end of processing of Phase-2. These audemes are dynamic in nature and can change according to the perception of listeners or users. This change is in the atomic-sounds being selected from the list, to portray a class for the concept. Also the system is capable of generating correct audeme (as perceived by the system with respect to its correct state, based on ranking of correlation factor) as well as random audemes (random mix of atomic-sounds irrespective of correlation factor). Following are some of the audemes generated by the system and are represented here as a sequence of name of sound files:

Fossil fuel:

Correct audeme: [ind_fire, ind_animal, ind_animal, A_clothing, ind_machine]

Random Audeme: [ind_fire, ind_earth, A_cyclicalChange, A_time, ind_machine]

Enzyme:

Correct audeme: [A_work, A_quick, A_cyclicalChange, A_structure, ind_machine]

Random Audeme: [A_food, A_structure, A_beginning, A_institution, ind_machine]

Again we can note from above sample outputs that some of the atomic-audemes are repeated in both the concepts. For example atomic-sound “ind_machine” is present in above mentioned audemes for “fossil fuel” and “enzyme”. This situation is because of fairly small number of atomic-sounds present in the system as of now.

The next part of the processing for Phase-3 is the processing for updating the correlation factor of atomic-sounds from the data collected by games/puzzles being played by users. We required random generation of audemes, so that the users can be given multiple options and if the audeme generated by the system is not best as per the user's perception then, the system will adapt to the user's result as described in section 3.5.

We created quiz for multiple science concepts and recorded the results from six users who played it (all the users in this study were sighted and provided with explanation of the concept). The quiz was to identify the best audeme for a concept out of the four options. For example the four audeme options given for adaptation were:

a) [ind_animal, ind_air, A_actions, ind_animal, A_time]
b) [ind_animals, ind_rising, A_below, A_peaceful, A_not]
c) [A_beginning, A_ending, A_sinking, A_peaceful, A_bad]
d) [A_interior, A_cyclicalChange, A_below, A_new, A_not]

Out of the six users: three selected option (a), two selected option (b), and one thought (c) is the best audeme for "adaptation". But as per the design of quiz audeme (a) was the correct audeme, calculated by system, on the basis of initial correlation factor. From the data that has been collected from players of quiz we can calculate the updated correlation factor and from the new calculations we calculated the new correlation factor for individual atomic-sounds in a class for a science-concept. In the above example the correlation factor of 'ind_animal' before integrating quiz data was '3' and after including the 'count' (calculated as described in section 3.5) from quiz results it is '8' which ranks 'ind_animal' as the most correlated "initiator" for "adaptation".

The interfaces for audeme interactions with visually impaired have already been used and tested for games [12]. Moreover new interfaces and games are being developed to further test usability of audemes and to accumulate feedback from users. One such interface has been developed on the Android phone platform, which provides a simple 'identify the audeme' puzzle with four options which can be played directly by a visually impaired student. These interfaces can provide direct feedback to the system and those feedback can be integrated into the system either real-time or after a specified time gap. Another interface is a web based interface wherein a student can interact with the audemes through an instructor.

5 CONCLUSION AND FUTURE WORK

5.1 Future Work

Audemes and their use as a supplement to pedagogy for visually impaired is still a new concept and more work needs to be done in spreading the idea. This will require generation of interfaces that increase the involvement of students in science and related fields. The future work for this system is to integrate it with external interfaces and allow automatic real-time update of correlation factor. The current implementation allows this to be done over a period of time. Although an android app is prepared for audeme usability but it has not been integrated with the system explained here. More and more interfaces can provide the required push for audeme and this may even help people with sight to understand science.

The proposed applications that can be built with audeme integration to latest hand-held devices can change the way we interact with concepts today. One of the proposed use interface can be a text reader integration, wherein a user will get the option (via pop-up link) to listen to the audeme of a known concept in the article being read.

The customization as is done in the current implementation is global and takes in consideration the cumulative feedback from all the users, but this can also be customized per user. That is the ranking of atomic-sounds per concept can be customized as per what a user want. This is particularly useful for instructors as they can customize audemes according to their personal way of explaining a concept in the classroom.

A small term future work to improve the performance of the system is the generation of more and more unique atomic-sounds and their integration in sound-word database. As we have seen in the previous chapter that final outcome of our system is limited because of a small number of atomic-words in sound-word database.

5.2 Conclusion

Since the usability of audemes in improving the education of Blind and Visually Impaired has already been proved [17]. The manual process of the creation of audeme was a bottle neck for actually using audemes for pedagogy with blind students. The methods presented in this thesis guarantee the quick supply of audemes and will help audeme games played as at blind schools to go along with the pace of the curriculum. Moreover since more and more blind schools are signing up for experiments with audemes, this automated system will be of much use to coordinate and supply audemes to multiple schools simultaneously. Also since there will be more users the system will evolve better using all the feedbacks from different resources.

The classification of events/actions/modifiers which define a science process into semantic categories provides a new way to look at science processes and can be used as a simple mechanism to understand different processes. This encapsulation of science process can be used in general for pedagogical purposes and a majority of processes can fit in to the definition of '*Initiator*', '*Condition*', '*Action*', '*Action-on*', '*Output*'.

In Phase-2 of processing for audeme generation the use of multiple resources ensure the robustness of the system. Synonyms fetched from thesaurus.com depends on the online API from dictionary.com, this limitation of being connected to a third party API is overcome by the parallel use of WordNet which is present locally on the system itself. An improvement can be made by fetching other related words like Hyponyms and Hypernyms from WordNet apart from just Synonyms; however this change can also lead to ambiguous results. The impact of adding other related words in the list while fetching synonyms can only be studied after series of experiments with actual users.

This system can be directly plugged into any platform that provides an interface for BVI students. The database can also be updated easily to change the correlation factor with the data collected from these multiple interfaces. The overall approach taken translation of a science concept from text to non-verbal sound uses manual intervention only in identifying the related things or actions while generating an atomic-sound for sound-word database. In any system the initial knowledge base is always created manually, moreover this information can be used for further work with non-verbal sounds.

LIST OF REFERENCES

LIST OF REFERENCES

[1] P.J. Tilgner. Avoiding science in the elementary school. *Science Education*, 74(4):421–431, 1990.

[2] S.W. Slough, E.M. McTigue, S. Kim, and S.K. Jennings. Science textbooks' use of graphical representation: A descriptive analysis of four sixth grade science texts. *Reading Psychology*, 31(3):301–325, 2010.

[3] I. Martins, J. Otero, JA Leon, and AC Graesser. Visual imagery in school science texts. *The psychology of science text comprehension*, pages 73–90, 2002.

[4] Wikipedia. Office Assistant. `http://en.wikipedia.org/wiki/A\_picture\_is\_worth\_a\_thousand\_words`.

[5] A.N. Hibbing and J.L. Rankin-Erickson. A picture is worth a thousand words: Using visual images to improve comprehension for middle school struggling readers. *The reading teacher*, 56(8):758–770, 2003.

[6] R. Lowe. Interrogation of a dynamic visualization during learning. *Learning and Instruction*, 14(3):257–274, 2004.

[7] J.K. Gilbert. The visualization of models: A metacognitive competence in the learning of chemistry. 2003.

[8] D. Hestenes. Modeling games in the newtonian world. *American Journal of Physics*, 60(8):732–748, 1992.

[9] G. Marbach-Ad, Y. Rotbain, and R. Stavy. Using computer animation and illustration activities to improve high school students' achievement in molecular genetics. *Journal of Research in Science Teaching*, 45(3):273–292, 2008.

[10] R. Tytler, J. Osborne, G. Williams, K. Tytler, and J. Cripps Clark. Opening up pathways: Engagement in stem across the primary-secondary school transition. 2012.

[11] National Federation of the Blind Jernigan Institute. Why is a national center for blind youth in science needed. 2010.

[12] M. Ferati, S. Mannheimer, and D. Bolchini. Usability evaluation of acoustic interfaces for the blind. In *Proceedings of the 29th ACM international conference on Design of communication*, pages 9–16. ACM, 2011.

[13] S. Mannheimer, M. Ferati, D. Bolchini, and M. Palakal. Educational sound symbols for the visually impaired. *Universal Access in Human-Computer Interaction. Addressing Diversity*, pages 106–115, 2009.

[14] M.C. De Marneffe, B. MacCartney, and C.D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[15] George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[16] M. Jarmasz and S. Szpakowicz. Rogets thesaurus: A lexical resource to treasure. In *Proceedings of the NAACL WordNet and Other Lexical Resources workshop*, pages 186–188. Citeseer, 2001.

[17] S. Mannheimer, M. Ferati, D. Huckleberry, and M. Palakal. Using audemes as a learning medium for the visually impaired. *Proceedings of HEALTHINF*, 9:175–180, 2009.

[18] M. Back and D. Des. Micro-narratives in sound design: Context, character, and caricature in waveform manipulation. In *Proceedings of the International Conference on Auditory Display*, 1996.

[19] Wikipedia. Office Assistant. `http://en.wikipedia.org/wiki/Earcon`.

[20] S.A. Brewster. *Providing a structured method for integrating non-speech audio into human-computer interfaces*. PhD thesis, PhD Thesis, University of York, UK, 1994.

[21] W. Buxton. Introduction to this special issue on nonspeech audio. *Human Computer Interaction*, 4(1):1–9, 1989.

[22] K.L. Stephan, S.E. Smith, R.L. Martin, S. Parker, and K.I. McAnally. Learning and retention of associations between auditory icons and denotative referents: Implications for the design of auditory warnings. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(2):288–299, 2006.

[23] P. Keller and C. Stevens. Meaning from environmental sounds: types of signal-referent relations and their effect on recognizing auditory icons. *Journal of experimental psychology: Applied*, 10(1):3, 2004.

[24] S. Koelsch, E. Kasper, D. Sammler, K. Schulze, T. Gunter, and A.D. Friederici. Music, language and meaning: brain signatures of semantic processing. *Nature neuroscience*, 7(3):302–307, 2004.

[25] V. Erlmann. *Hearing cultures: essays on sound, listening, and modernity*. Berg Publishers, 2004.

[26] A.D.N. Edwards. Modelling blind users' interactions with an auditory computer interface. *International Journal of Man-Machine Studies*, 30(5):575–589, 1989.

[27] W.W. Gaver. The sonicfinder: An interface that uses auditory icons. *Human-Computer Interaction*, 4(1):67–94, 1989.

[28] J. Sánchez and H. Flores. Memory enhancement through audio. *ACM SIGACCESS Accessibility and Computing*, (77-78):24–31, 2003.

[29] M.E. Doucet, J.P. Guillemot, M. Lassonde, J.P. Gagné, C. Leclerc, and F. Lepore. Blind subjects process auditory spectral cues more efficiently than sighted individuals. *Experimental brain research*, 160(2):194–202, 2005.

[30] J. Sánchez, M. Lumbreras, and L. Cernuzzi. Interactive virtual acoustic environments for blind children: computing, usability, and cognition. In *CHI'01 extended abstracts on Human factors in computing systems*, pages 65–66. ACM, 2001.

[31] M. Ferati, S. Mannheimer, and D. Bolchini. Acoustic interaction design through audemes: experiences with the blind. In *Proceedings of the 27th ACM international conference on Design of communication*, pages 23–28. ACM, 2009.

[32] G. Roma, J. Janer, S. Kersten, M. Schirosa, and P. Herrera. Content-based retrieval from unstructured audio databases using an ecological acoustics taxonomy. 2010.

[33] Merine. Thomas. Identification of directional relations between biological objects from biomedical literature. Master's thesis, Masters Thesis, Purdue University, USA, 2004.

[34] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates. Web-scale information extraction in knowitall:(preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, pages 100–110. ACM, 2004.

[35] C.H. Chang, M. Kayed, R. Girgis, and K.F. Shaalan. A survey of web information extraction systems. *Knowledge and Data Engineering, IEEE Transactions on*, 18(10):1411–1428, 2006.

[36] A.H.F. Laender, B.A. Ribeiro-Neto, A.S. Da Silva, and J.S. Teixeira. A brief survey of web data extraction tools. *ACM Sigmod Record*, 31(2):84–93, 2002.

[37] S. Sarawagi. Information extraction. *Foundations and trends in databases*, 1(3):261–377, 2008.

[38] B.B. Chaudhuri. *Digital document processing: major directions and recent advances*. Springer-Verlag New York Inc, 2007.

[39] Wikipedia. Office Assistant. `http://en.wikipedia.org/wiki/Grep`.

[40] T.K. Ho. Fast identification of stop words for font learning and keyword spotting. In *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on*, pages 333–336. IEEE, 1999.

[41] R. Al-Shalabi, G. Kanaan, J.M. Jaam, A. Hasnah, and E. Hilat. Stop-word removal algorithm for arabic language. In *Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on*, page 545, april 2004.

[42] Lili Hao and Lizhu Hao. Automatic identification of stop words in chinese text classification. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 1, pages 718 –722, dec. 2008.

[43] J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. Indexing with wordnet synsets can improve text retrieval. *Arxiv preprint cmp-lg/9808002*, 1998.

[44] Jonathan J. Webster and Chunyu Kit. Tokenization as the initial phase in nlp. In *Proceedings of the 14th conference on Computational linguistics - Volume 4*, COLING '92, pages 1106–1110, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[45] L. Karttunen, J.P. Chanod, G. Grefenstette, and A. Schille. Regular expressions for language engineering. *Natural Language Engineering*, 2(04):305–328, 1996.

[46] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O'Reilly Media, 2009.

[47] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the workshop on Speech and Natural Language*, HLT '91, pages 112–116, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[48] Thorsten Brants. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[49] J. Kupiec. Robust part-of-speech tagging using a hidden markov model. *Computer Speech & Language*, 6(3):225–242, 1992.

[50] K. Toutanova, D. Klein, C.D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[51] M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177. Association for Computational Linguistics, 2008.

[52] A. Björkelund, B. Bohnet, L. Hafdell, and P. Nugues. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, pages 33–36. Association for Computational Linguistics, 2010.

[53] D.A. Hull. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1):70–84, 1996.

[54] P. Willett. The porter stemming algorithm: then and now. *Program: electronic library and information systems*, 40(3):219–223, 2006.

[55] R. Chapman. *Rogets International Thesaurus (Fifth Edition)*. Harper Collins, 1992.

[56] M. Jarmasz and S. Szpakowicz. Rogets thesaurus and semantic similarity1. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 2003:111, 2004.

[57] S.S. Lee and H.S. Yong. Tagplus: A retrieval system using synonym tag in folksonomy. In *Multimedia and Ubiquitous Engineering, 2007. MUE'07. International Conference on*, pages 294–298. IEEE, 2007.

[58] L. Lan. The growing prosperity of on-line dictionaries. *English today*, 21(3):16–21, 2005.

[59] S. Marsland. *Machine learning: an algorithmic perspective*. Chapman & Hall/CRC, 2009.

[60] M.C. De Marneffe and C.D. Manning. Stanford typed dependencies manual. *URL http://nlp. stanford. edu/software/dependencies_manual. pdf*, 2008.

[61] D. Klein and C.D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

[62] Wikipedia. Office Assistant. `http://en.wikipedia.org/wiki/Onomatopoeia`.

APPENDIX

APPENDIX

initial state	dependency	final state	pos
1	prep_as	2	
1	nsubj	1	
1	amod	1	
1	nn	1	
1	num	1	
1	prep_of	1	
1	prep_such_as	2	
1	advmod	1	
1	rcmod	2	
1	prep_is	2	
1	partmod	4	
1	prep_including	2	
1	prep_in	2	
2	nsubj	1	
2	dobj	2	
2	prep_in	2	
2	prep_into	2	
2	prep_or	2	
2	prep_to	2	
2	conj_or	2	
2	prep_through	2	

2	amod	2	
2	nn	2	
2	agent	2	
2	num	2	
2	rcmod	4	
2	partmod	4	
2	xcomp	4	
2	infmod	4	
2	cop	2	
2	advcl	4	V
2	prep_of	2	
3	prep_in	3	
3	nsubj	5	
3	rcmod	2	
3	partmod	4	
3	prep_from	2	
3	prep_to	2	
3	prep_at	2	
3	nn	3	
3	amod	3	
3	prep_such_as	2	
3	prep_of	3	
3	dobj	3	
4	prep_along_with	5	
4	dobj	3	
4	prep_over	3	
4	appos	5	
4	prep_as	4	
4	prep_to	3	

	4	prep_on	3	
	4	nsubjpass	3	
	4	prep_with	2	
	4	xsubj	1	
	4	prep_of	3	
	4	prep_against	3	
	4	ccomp	4	
	4	nsubj	1	
	4	agent	1	
	4	advmod	4	
	4	prep_during	2	
	4	prep_in	2	
	4	prep_for	5	
	4	prep_by	1	
	4	prep_at	2	
	4	advcl	4	V
	4	xcomp	4	
	4	conj_and	4	
	4	aux_pass	4	
	4	prep_through	2	
	4	prt	4	
	4	aux	4	
	5	npadvmod	5	
	5	prep_like	5	
	5	nn	5	
	5	amod	5	
	5	conj_and	5	
	5	prep_of	5	
	5	poss	5	

5	det	5	
5	num	5	
5	prep_as	5	
5	xcomp	4	
5	rcmod	3	
5	partmod	4	
5	prep_in	5	
2	advmod	2	
1	conj_or	1	
2	conj_and	2	N
4	auxpass	4	
5	prep_with	2	
5	predet	5	
2	predet	2	
1	predet	1	
4	mark	4	
3	nsubjpass	3	
3	complm	3	
3	auxpass	3	
2	parataxis	4	
1	appos	1	
2	prep_off	2	
2	prep_such_as	2	
5	prep_such_as	2	
4	prep_into	2	
2	prep_for	2	
4	prep_from	2	
2	prep_over	2	
2	auxpass	2	

4	pcomp	4	
1	prep_from	2	
2	prep_from	1	
2	aux	2	
3	conj_or	3	
4	conj_or	4	
1	conj_and	1	
5	prep_from	2	
3	num	3	
2	prep_as	2	
5	quantmod	5	
2	quantmod	2	
1	quantmod	1	
1	prep_to	1	
5	prep_over	5	
2	mwe	2	
2	poss	2	
2	prep_like	2	
1	prep_over	2	
3	aux	4	
5	prep_or	5	
4	neg	4	
5	conj_or	5	
4	acomp	5	
2	advcl	2	N
4	advcl	2	N
5	advmod	5	
3	advmod	3	
4	amod	4	

	1	agent	1	
	5	agent	5	
	2	appos	2	
	3	apppo	3	
	5	appos	5	
	5	attr	5	
	1	aux	1	
	5	aux	5	
	1	auxpass	1	
	5	auxpass	5	
	1	ccomp	4	
	2	ccomp	4	
	3	ccomp	4	
	5	ccomp	4	
	3	conj_and	3	
	1	conj_but	2	
	2	conj_but	2	
	4	conj_but	4	
	5	conj_but	5	
	5	cop	4	
	1	cop	1	
	4	cop	4	
	3	cop	4	
	2	csubj	4	
	4	csubj	4	
	5	csubj	4	
	5	dep	4	V
	1	dep	4	V
	2	dep	4	V

	3	dep	4	V
	4	dep	4	V
	1	dobj	1	
	5	dobj	5	
	2	neg	2	
	1	neg	1	
	5	neg	5	
	3	neg	3	
	1	attr	1	
	2	attr	2	
	3	attr	3	
	4	attr	4	
	1	cmod	2	
	4	comp	5	
	1	conj_plus	1	
	2	conj_plus	2	
	3	conj_plus	3	
	4	conj_plus	4	
	5	conj_plus	5	
	1	det	1	
	2	det	2	
	3	det	3	
	4	det	4	
	1	infmod	4	
	3	infmod	4	
	4	infmod	4	
	4	csubjpass	4	
	4	iobj	3	
	4	ibj	4	

2	mark	2	
3	mark	3	
1	mark	2	
1	mwe	1	
4	nn	1	
5	nsubj	1	
2	nsubjpass	1	
1	nsubjpass	1	
5	nsubjpass	5	
2	number	2	
3	number	3	
1	number	1	
4	number	4	
5	number	5	
1	parataxi	4	
1	dep	2	N
2	dep	2	N
3	dep	2	N
4	dep	2	N
5	dep	2	N
4	num	4	
4	pobj	1	
2	pobj	2	
5	pobj	2	
4	poss	3	
3	poss	3	
1	poss	1	
4	prep_after	2	
3	prep_after	1	

2	prep_after	2	
5	prep_after	1	
2	promp	2	
2	preconj	2	
1	prep	1	
2	prep	2	
3	prep	3	
4	prep	4	
5	prep	5	
1	prep_about	2	
3	prep_about	2	
4	prep_above	2	
1	prep_across	1	
3	prep_against	2	
5	prep_against	2	
1	prep_around	2	
2	prep_around	2	
3	prep_around	2	
4	prep_around	2	
5	prep_around	2	
3	prep_as	2	
1	prep_at	2	
2	prep_at	2	
5	prep_at	5	
4	prep_between	2	
1	prep_between	2	
2	prep_between	2	
3	prep_between	2	
5	prep_between	2	

2	prep_by	1	
1	prep_by	1	
5	prep_by	3	
2	prep_close_to	1	
2	prep_during	2	
1	prep_during	2	
3	prep_for	2	
4	prepc_by	4	
5	prep_for	2	
3	cmod	2	
2	cmod	4	
5	cmod	4	
4	rel	3	
3	rel	4	
2	rep_as	2	
3	rep_of	3	
1	rtmod	4	
5	subj	5	
4	subj	3	
4	tmod	5	
1	tmod	2	
2	tmod	2	
1	xsubj	1	
4	prep_including	3	
3	prep_including	2	
3	prep_inside	3	
3	prep_into	2	
2	prt	2	
4	prep_unlike	2	

4	prep_behind	3	
2	prepc_by	2	
4	complm	4	
2	prep_with	2	
1	prep_with	2	
4	prep_across	2	
1	prep_for	2	
2	prep_on	2	
4	rcmod	4	
2	prep_despite	2	
4	prep_despie	2	
5	infmod	4	
5	prep_on	2	
4	prep_throughout	3	
1	prep_against	2	
4	prepc_as	4	
3	prepc_as	2	
1	prepc_as	5	
5	prepc_as	5	
2	prepc_as	2	
5	prep_to	2	
4	prep_than	2	
3	prep_than	2	
2	prep_than	2	
1	prep_than	2	
5	prep_than	2	
4	prepc_on	4	
2	prepc_according_to	2	
4	prepc_according_to	2	

5	prepc_according_to	2	
2	prep_before	2	
1	prep_near	2	
2	prep_near	2	
3	prep_near	2	
4	prep_near	2	
5	prep_near	2	
4	prep_though	2	
3	prep_though	2	
2	prep_though	1	
1	prep_though	2	
5	prep_through	2	
5	prep_though	2	
2	rel	3	
4	prepc_instead_of	4	
2	iobj	4	
2	prepc_in	4	
1	prepc_of	4	
2	prepc_of	4	
3	prepc_of	4	
4	prepc_of	4	
5	prepc_of	4	
3	prep_through	2	
3	xcomp	4	
4	prepc_in	4	
5	prepc_in	2	
3	agent	3	
4	partmod	4	
1	prep_on	2	

	4	prep_out_of	2	
	3	prep_over	2	
	4	prep_since	2	
	3	prep_throughout	2	
	1	prep_throughout	2	
	2	prep_toward	2	
	1	prep_toward	1	
	4	prep_under	2	
	3	prep_with	2	
	4	prep_like	2	
	3	prep_like	2	
	1	prep_like	2	
	4	prep_within	1	
	1	prep_within	2	
	4	prep_without	2	
	4	prep_according_to	2	
	5	prep_according_to	2	
	4	prepc_after	2	
	5	prepc_by	4	
	4	prepc_from	4	
	2	prepc_from	3	
	1	prt	1	
	3	prt	3	
	5	prt	5	
	3	quantmod	3	
	4	quantmod	4	
	2	pcomp	2	
	4	subjpass	1	
	5	prep_throughout	2	

2	prep_throughout	1	
1	xcomp	4	
4	prep_despite	2	
3	prep_despite	2	
1	prep_despite	2	
5	prep_despite	2	
5	advcl	4	
4	prep_away_from	3	
1	papataxis	4	
2	npadvmod	2	
4	npadvmod	2	
4	prep_toward	2	
5	comp	5	
4	prep_because_of	2	
2	prep_because_of	1	
3	prep_because_of	2	
5	prep_because_of	4	
1	prep_because_of	2	
4	prep_below	2	
1	prep_below	2	
3	prep_below	2	
2	prep_below	2	
1	prep_according_to	2	
2	prep_according_to	2	
3	prep_according_to	2	
1	pobj	2	
1	prep_through	2	
1	csubj	4	
3	csubj	4	

1	conj	4	
3	conj	2	
1	parataxis	4	
3	parataxis	4	
5	parataxis	4	
1	advcl	4	V
1	advcl	2	N
1	prepc_according_to	2	
3	prepc_according_to	2	