The structure of conversation
real or random?

Lydia Behan
B.A. (Mod.) CSLL
Final Year Project, April 2010
Supervisor: Dr. Carl Vogel
Declaration

I hereby declare that this thesis is entirely my own work and that it has not been submitted as an exercise for a degree at any other university.

_______________________________________  April 7, 2010
Lydia Behan
Permission to Lend

I agree that the Library and other agents of the College may lend or copy this thesis upon request.

_________________________________________ April 7, 2010
Lydia Behan
Acknowledgments

I wish to thank the following people:
Dr. Carl Vogel, my supervisor, for all of his advice and help during this project and over the last four years.
Dr. Nick Campbell for allowing me to use his corpus of transcripts for this project.
My fellow CSLL students for their encouragement and friendship.
Last but by no means least, I would like to thank my family for their unending support over the last four years.
Abstract

This paper aims to prove that natural conversation is structured by demonstrating that features of conversation such as repetition, overlap and priming do not occur randomly in conversation. The approach taken in this project is to compare the real order and shuffled order of conversation using the purpose-designed automated analysis program created for this project. The results suggest that these features appear to be orderly, demonstrating that conversation has an underlying structure.
## Contents

1 Introduction ........................................... 10

2 Background material .................................. 11
   2.1 Introduction ...................................... 11
   2.2 Linguistic Priming ................................ 13
      2.2.1 The lexical decision task .................. 15
      2.2.2 The confederate scripting experiment ...... 15
   2.3 Repetition in conversation ....................... 17
      2.3.1 Repetition in the early years ............... 17
      2.3.2 Functions of repetition in child language ... 17
      2.3.3 Repetition in adult speech ................. 17
      2.3.4 Functions of repetition in conversation .... 18
   2.4 The organisation of conversation ............... 25
      2.4.1 Turn Taking .................................. 25
      2.4.2 Overlap ..................................... 27
      2.4.3 Gender and turn-taking ..................... 32
   2.5 Film dialogue .................................... 34
   2.6 Conclusion ...................................... 35

3 Main experiment ...................................... 36
   3.1 Introduction ...................................... 36
   3.2 Aims ............................................. 36
   3.3 Transcripts ...................................... 37
      3.3.1 Choosing a corpus ........................... 37
      3.3.2 Corpus used .................................. 38
      3.3.3 Changes made to the transcripts ............ 39
   3.4 Participants ..................................... 39
   3.5 Conversations .................................... 40
   3.6 Programming language used ..................... 41
   3.7 Program design .................................. 41
      3.7.1 Shuffled Transcripts ......................... 41
4 Results
4.1 Introduction ................................................... 49
4.2 Repetition ..................................................... 49
  4.2.1 Day 1 ....................................................... 49
  4.2.2 Day 2 ....................................................... 51
  4.2.3 Day 3 ....................................................... 52
  4.2.4 Average repetition ......................................... 54
  4.2.5 Real and fake overlap ..................................... 57
  4.2.6 Day 1 ....................................................... 59
  4.2.7 Day 2 ....................................................... 60
  4.2.8 Day 3 ....................................................... 60
  4.2.9 Average overlap ........................................... 61

5 Discussion and Analysis of results 64
5.1 Introduction ................................................... 64
5.2 Discussion and Analysis ........................................ 64
  5.2.1 Repetition ................................................... 64
  5.2.2 Overlap .................................................... 67
  5.2.3 Repeated overlap ......................................... 68
5.3 Conclusion .................................................... 68

6 Future research ................................................ 69
  6.1 Continuation of this project ................................. 69
  6.2 Artificial Intelligence ........................................ 69
  6.3 Film and television .......................................... 70

7 Conclusion ..................................................... 71

A Transcription conventions 72
List of Figures

1. Occurrences of repetition on day 1 for \( n = 1 \) to \( n = 5 \) ........... 50
2. Occurrences of repetition on day 2 for \( n = 1 \) to \( n = 5 \) ........... 51
3. Occurrences of repetition on day 3 for \( n = 1 \) to \( n = 5 \) ........... 53
4. The average number of occurrences of repetition over 3 days for \( n = 1 \) to \( n = 5 \) ........................................ 54
5. The effect of \( n \) and day on repetition ........................................ 55
6. The occurrences of overlap by day .............................................. 57
7. The effect of day and next on fake overlap ................................. 58
8. The occurrences of repeated overlap for day 1 ............................. 59
9. The occurrences of repeated overlap on day 2 ............................. 60
10. The occurrences of repeated overlap on day 3 ............................. 61
11. The average number of occurrences of repeated overlap ............. 62
12. The effects of reality, \( n \) and day on fake repeated overlap ....... 63
1 Introduction

During the course of looking at the history of research into the structure of conversation, it became clear that the research reviewed was based on conversation in its natural order but the research did not explore alternative ordering.

This project compares the real order and shuffled order of conversation with the purpose of exploring the features of natural conversation from a comparative viewpoint. Three features of natural conversation are examined in this research:

- priming
  - syntactic priming
  - semantic priming
  - lexical priming
- repetition
  - self-repetition
  - other speaker repetition
- overlap
  - back-channelling
  - interruption
  - simultaneous start-ups

All of the original research reviewed during this project had simply examined these features by hand. In order to examine these features in this project, an automated analysis program was created. The various factors which affect these features in conversation were also discussed.

The aim of this project is to prove the theories of conversation structure which are introduced in section 2 by comparing ordered and unordered conversations in an automatic fashion. The creation of this analysis program is discussed, as well as the corpus of conversations which it analysed. The results of this analysis program are shown, followed by a discussion and interpretation of these results. Finally, future uses of the results of this project are explored.
2 Background material

2.1 Introduction

Conversation seems to be a highly disorganised form of communication. Yet if this is true, how do we manage to converse with others on a daily basis? In the words of Garrod and Pickering, why is conversation so easy? The answer to this question may lie in the fact that conversation has structure. These “rules” aren’t always explicit yet we seem to know intuitively when someone has broken them. Phrases like “he talks too much”, “don’t interrup”, “you’re repeating yourself” and “I can’t get a word in edgewise” exist because we feel that other speakers are breaking the rules of conversation, and as these rules are not explicit, we need to make it clear to others that we feel they have violated them. These rules are, however, not strict or constraining:

“Rather, they are rules in that their conduct people display an orientation to. There is a distinction then between rules that people can be shown to orient to and rules that are said to be an interior mental machinery. On the latter understanding, rules stand behind action, on the former, rules are embedded within the action.”
(Button quoted in ten Have [1999])

These rules show that there is a structure to conversation. There are rules concerning turn-taking which help us to decide who should speak next, and when it is time to take up our turn. The utterances which we produce during a turn are not random either. People tend to continue on the same subject as the previous speaker, unless they have deliberately (and obviously) changed the subject. This follows Grice’s cooperative principle:

“make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged” [Grice, 1989]

Yet how do we achieve this task of staying on topic? The phenomenon of priming explains how speakers are influenced by preceding utterances. This may go some way to explaining how we keep on topic. How do we act as cooperative
speakers? Garrod and Pickering suggest that the act of conversation is a joint activity in which speakers “work together to establish a joint understanding of what they are talking about”. They do this through “interactive alignment”. Interactive alignment is the process during which both (or all) speakers align their mental representations at different linguistic levels [Garrod and Pickering, 2004]. How do speakers go about this alignment? Garrod and Pickering state that interactive alignment comes about for two reasons:

- Parity of representations used in production and comprehension
- Priming of representations between speakers and listeners

Repetition is another way in which we achieve alignment in conversation. Repetition can be used by speakers for clarification and to repair utterances which may have contained errors. This confirmation and error correction aids the alignment process as there is a clear understanding between speakers.

The ideas of turn-taking, priming and repetition are explained to see the effect these phenomena have on conversation. Also how these phenomena contribute to the idea that conversation is highly structured is shown.
2.2 Linguistic Priming

Baars describes priming as being when one experience affects the interpretation of another experience [Baars, 1997]. Priming is a non-conscious form of human memory [Tulving and Schacter, 1990]. For psychologists, priming experiments provide an interesting insight into the workings of the mind. Results from such experiments have given psychologists a better understanding of how humans store the vast quantities of information we accumulate throughout our lifetimes and how we retrieve these items from memory.

For linguists, priming experiments give a broader understanding of how humans organise language. How do we understand and interpret utterances by others given the numerous possible interpretations available to us? How do we understand words which have multiple meanings? Baars explains that it is priming which allows us to achieve this incredible task.[Baars, 1997] He uses word pairs to illustrate this:

\[
\begin{align*}
\text{pages} & : \text{book} \\
\text{arrest} & : \text{book}
\end{align*}
\]

We interpret the word “book” differently for each word pair and it is the preceding prime which allows us to do so. We are influenced by the prime in our understanding of the following word. Does our interpretation of “book” now have an effect on words which follow it? Swinney [1979] carried out an experiment to investigate this matter. In this experiment, participants were presented with sentences such as the following:

“Rumour had it that, for years, the government building had been plagued with problems. The man was not surprised when he found several bugs in the corner of his room”

The sentence above was an example of an ambiguous sentence with no context. Three other types of sentences were given: Ambiguous sentences with biasing con-
text ("bugs" was replaced in the above sentence with "several spiders, roaches, and other bugs"), unambiguous sentences with no context ("bugs" was replaced with "several insects") and unambiguous sentences with biasing context ("bugs" was replaced with "several spiders, roaches, and other insects") [Swinney, 1979]. Each participant was played the sentences through headphones and asked to listen to the sentence and to understand it (Participants were also given a comprehension test to ensure that they had understood the sentences which were played to them). The participants were then immediately given the lexical decision task (see section 2.2.1) The results of the task showed that lexical decisions based on both interpretations of the ambiguous word were facilitated in sentences with biasing and non-biasing contexts. Participants decided on words such as "insect" and "spy" with similar speeds in comparison to non-words. However, when the lexical decision task was delayed, such results were not seen. The biasing context factor came into account when there was a time delay between the sentences being presented and the lexical decision task being taken. This delay is known as the stimulus onset asynchrony or SOA. The results of this experiment show that although the initial retrieval of a word’s meaning was not dependent on the context surrounding the word, only the contextually appropriate meaning of the word contributes to how the sentence meaning is retained in our memory [McNamara, 2005].

According to Hoey, the priming of a word is not permanent. Encounters with primes either reinforce or undermine word associations. This means that primings are strengthened or weakened every time we encounter them. Context also has a huge effect on priming, as was seen above. Changes in primings over time can also contribute to language change within a language community.

Another reason for the priming of a word to change is through the phenomenon of cracks. Cracks are when the speaker is advised that the priming of a word is incorrect. This generally takes place in the education setting. For example, a student would incorrectly form the sentence you + to be (past tense) as "you was". The teacher would inform the student that the correct structure is in fact "you were". The student has three ways to repair this crack. He can choose to accept the new prime and reject the old prime, he can reject the new prime and
continue to use the old prime, or he can use each prime in different settings, using the old prime in the home setting and the new prime in school. If a crack is not repaired, it can lead to long-term linguistic insecurity as the speaker will always be uncertain of the priming of that word [Hoey, 2005].

2.2.1 The lexical decision task

The lexical decision task is one of the most commonly used experiments in the examination of linguistic priming. In the lexical decision task, participants are given a pair of words, the prime and the target. Participants are shown the prime. Then they are shown the target and asked to decide whether or not the target is a word or a non-word (meaningless strings of letters). Experimenters have found that participants respond more quickly and more accurately to target words which were semantically related to the prime. This means that the word “nurse” would be found to be a word more quickly if it had been preceded by “doctor” than by “butter”. The relationship between the target and the prime can also be associative. The word “dog” would be recognised as a word more quickly if preceded by “cat” than by “goat”. Although “goat” and “dog” are semantically related, they are rarely associated with each other, unlike “cat” and “dog”.

Word frequency also has a strong effect on the recognition of words. Targets such as “year” are recognised as words more quickly and more accurately than uncommon words such as “hermeneutic” [Harley, 2001]. The plausibility of the non-word also has an effect on the time taken to reject it as a word. For instance, “fead” takes longer to reject than “thnze”. “fead” has a similar structure to English words such as “bead” and “head”, whereas “thnze” does not follow any of the phonological rules of English. It is for this reason that “thnze” will be more quickly and accurately rejected than “fead”.

2.2.2 The confederate scripting experiment

Branigan et al. [2004] conducted an experiment called the confederate scripting experiment in order to examine syntactic priming in dialogue. Two participants (one is a confederate participant and one is a genuine subject) are asked to sit on opposite sides of a table with a divider in between them. The participants cannot
see each other throughout the experiment. Each participant is given a pile of cards and asked to describe their card to the other participant so that they can find the corresponding card in their pile. The confederate is given a scripted sentence for each card that they have to describe to the subject. The scripted sentences given to the confederate belong to two categories: DO (eg. The chef giving the jug to the swimmer) and PO sentences (eg. The chef giving the swimmer the jug) [Branigan et al., 2004]. The confederate describes their card to the subject and then the subject describes their card to the confederate. This order is always maintained as the prime must come before the target. Branigan et al. [2004] found that subjects regularly repeated the syntactic structure of the prime when it was their turn to present a card. This experiment showed that speakers are not only influenced by the individual words of other speakers, but also the syntactic structure of their utterances.
2.3 Repetition in conversation

2.3.1 Repetition in the early years

Repetition becomes an aspect of our linguistic style in infancy, before we could be said to be truly talking. Between the ages of 7-10 months, babies babble in a manner known as reduplicated or syllabic babbling. [Petitto and Marentette, 1999] This is when babies utter syllables repeatedly such as “babababa” and “dadada”. Petitto and Marentette (ibid) say that although babbling was thought to be the result of the maturation of the speech organs, observation of deaf babies has revealed that those babies who are exposed to sign language (or baby sign) babble using their hands (manual babbling). This, the authors say, proves that babbling is not simply the product of the maturation of the vocal system of a child, but also due to the maturation of the brain taking place at the same time.

Repetition does not end with the advent of speech. Repetition is of major importance in the acquisition of language. One of the main features of child directed speech (more commonly known as “baby talk”) is the repetition and shortening of syllables, words and utterances. In CDS, one might find “choo choo” instead of “train” and “din din” instead of “dinner”. Snow [1972], among others, has shown that parents talk differently to their young children than to their older children, using CDS with the former, however the benefits of such speech are debatable.

2.3.2 Functions of repetition in child language

Snow found that the value of this repetitive type of speech is that it allows children more time to process utterances. Paraphrasing allows young children to understand utterances in which there was vocabulary that they did not understand at the first statement.

2.3.3 Repetition in adult speech

Repetition continues to be a feature into adult speech. Speakers often repeat words, ideas or even whole utterances spoken by either themselves or another speaker. Tannen [2007] classifies repetition by several criteria: she distinguishes
self-repetition from allo-repetition (repetition of others) and also fixity of form. This fixity of form ranges from exact repetition to paraphrasing. The midpoint on her fixity scale is repetition with variation. This type of repetition is when a speaker repeats a word or phrase but changes the form of the original utterance, such as changing a statement into a question or changing person or verb tense. However, repetition is often viewed negatively. Many linguists consider that repetitions were markers of sloppy or disfluent speakers [Wong, 2000]. Tannen notes that repetition is not desirable in conversation, noting that the phrase “you’re repeating yourself” only has negative interpretations. Yet Bolinger makes the point that most of language is “pre-packaged”:

“At present we have no way of telling the extent to which a sentence like I went home is a result of invention, and the extent to which it is a result of repetition, countless speakers before us having already said it and transmitted it to us in toto. Is grammar something where speakers “produce” (i.e. originate) constructions, or where they “reach for” them, from a pre-established inventory...?” [Bolinger quoted in Tannen [2007]]

Tannen and Schegloff, amongst others, felt that these utterances should be examined due to their frequency among child and adult speakers alike and the fact that they seemed to be part of everyday conversation implied that they were not the characteristics of disfluent speakers [Wong, 2000].

2.3.4 Functions of repetition in conversation

If it is agreed that repetition is a feature of conversation, what is its purpose? The following are examples of the function of repetition in conversation:

1. Participatory listenership
2. Confirmation and clarification
3. Self repair and stalling
4. Effect
5. Truth

Each of these functions of repetition will now be discussed.

1. **Participatory listenership**

   In this type of repetition, speakers often repeat another speaker’s utterance in order to maintain their involvement in the conversation. They repeat the speaker word for word and do not add any new information to the conversation. She gives the following extract from a conversation:

   i. DEBORAH You know who else talks about that?
   ii. DEBORAH Did you ever read R.D. Laing?
   iii. CHAD The Divided Self?
   iv. CHAD Yeah. But I don’t /??/  
   v. DEBORAH He talks about that too
   vi. CHAD He talks about it too

   [Tannen, 2007]

   Tannen explains that the purpose of Chad’s input on line vi is to show acceptance of Deborah’s utterance and to show that he is listening. This type of repetition allows Chad to show Deborah that he is passively participating in the conversation and that although he accepts what she has to say, he has nothing to add to her utterance.

2. **Confirmation and clarification**

   Perrin et al. state that repetition can be used as a confirmation request. A speaker repeats the utterance of the previous speaker in order to signal a potential problem in the utterance and to initiate a repair [Perrin et al., 2003]. The speaker may feel that they have misunderstood or misheard the previous discourse and want the other speaker to clarify what they said.

   Father: Il s’est fait tuer (He’s been killed)
Mother: Il s’est fait tuer? (He’s been killed?)
Father: Oui il est arrivé un accident puis le chien s’est fait tuer. (Yes there was an accident and the dog’s been killed)
[Perrin et al., 2003]

In the above example, the mother has repeated the utterance of the father in order to confirm the validity of what he has said. The father understands her utterance as being a request for confirmation, as is illustrated by his response.

3. Self repair and stalling

Rieger [2003] suggests that repetitions can be the sign of self-repair. She describes self-repair as “error-correction, the search for a word,, and the use of hesitation pauses, lexical, quasi-lexical, or non-lexical pause fillers, immediate lexical changes, false starts, and instantaneous repetitions.” Rieger’s study was based on the analysis of conversations between English-German bilinguals. She found that although bilinguals use repetitions as self-repairs strategies, they do so differently depending on the language that they are speaking at the time. Different word categories are repeated depending on the language being spoken. Her analysis revealed that the most common functions for repetitions as self-repair were to delay the production of the next lexical item (ie. In order to allow the speaker time to search for the next word) and to hold the floor during what she calls a TRP (transition-relevance place). (see section 2.4)

Stalling using the mechanism of repetition can allow the speaker to fill in a pause in conversation while they search for their response. Tannen uses an example of two speakers talking on the subject of American Sign Language.

Peter: But how do you learn a new sign?
David: . . . How do I learn a new sign?
[Tannen, 2007]

In repeating Peter’s utterance, David has slowed the conversation and given
himself time to produce a response to Peter’s question. This simple use of repetition has allowed David to control the speed of the conversation and limit any pauses.

This stalling mechanism can be effective in an interview setting where one generally wants to give a good impression to the interviewer with good responses to their questions. Repetition of an utterance can allow the interviewee time to produce an answer without leaving pauses in the conversation which might indicate that they do not know the answer to the question.

Interviewer: Là ça fait combien de temps que vous faites ce nouveau travail là? (translation: How long have you had this new job?)
Interviewee: Le nouveau travail?: ça fait un mois là seulement (translation: The new job?: it’s only been a month)
[Perrin et al., 2003]

4. Repetition for effect

Repetition, although a common occurrence in natural conversation, is subconscious. We don’t plan to repeat ourselves or others. This is largely due to the fact that conversation is not itself planned. What do we observe when we look at planned conversation, such as in films or on television shows, during prewritten speeches, and in poetry?

- Conversations in films and television programs
  Repetition in films and television programs has two purposes. The first is to mimic natural conversations and the second is to create a desired effect. It is the second which is the primary purpose [Kozloff, 2000]. Kozloff uses an example taken from the film Good Will Hunting (1997) to show how repetition is used for effect. The main character, Will, is speaking to a psychiatrist, Sean. Sean attempts to comfort Will about the physical abuse Will suffered as a child. He says “it’s not your fault” and then repeats this phase half a dozen times. The purpose of the repetition according to Kozloff is that “with each repetition he penetrates further through Will’s defenses and pain”. Another common
pattern of repetition in films is when a character repeats one phrase many times throughout the film. Kozloff quotes John Fawell as saying that “the most memorable lines in the film are simple ones that are repeated”. Many examples of this type of repetition are given by Kozloff.

From Gone with the Wind (1938), Scarlett’s method of dealing with troubles: “I’ll think about it tomorrow”

From The Godfather (1972), Don Corleone’s business methodology: “I’ll make him an offer he can’t refuse” [Kozloff, 2000]

The effect of these repetitions is clear. The viewer becomes familiar with the line and it takes on a more powerful meaning with each repetition [Kozloff, 2000].

Repetition can also be used for humour. In this scene taken from Desperate Housewives, Susan introduces her new friend, Robin, to Roy and Karen (Roy’s partner).

Susan: This is my friend Robin
Roy (to Robin): Hello there
Susan: Mike had to work today so I thought I’d bring her as my plus one
Roy (to Robin): Hello there
Karen: You already said that
Roy: I wanted her to know I meant it

• Public speaking
A major part of the work of politicians is public speaking. There are some notable differences between natural speech and a prepared speech. In natural speech, as seen above, repetitions of words and phrases are common. In planned speech, unnecessary repetitions should be avoided. Repetition is used for effect and emphasis in planned speech.
The purpose of public speaking is to inform or influence an audience. As such, speakers should deliver a speech in a clear and concise method. If delivered correctly, a speech can be very moving and powerful. Many politicians are known for their public speaking skills. The effect of the “I have a dream” speech by Martin Luther King lives on almost forty years later.

- Poetry and song

Nowhere else is repetition used for effect as much as in poetry and song. Repetition of words, phrases, sentence structures and sounds are all seen in both of these artistic uses of language. A common repetition pattern seen in poetry is that of alliteration. Alliteration is when the same consonants are used in a line or verse of a poem. Another similar pattern is that of assonance, when vowel sounds are echoed. An excellent example of repetition in poetry can be seen in Edgar Allen Poe’s The Raven:

While I nodded, nearly napping, suddenly there came a tapping,
As of some one gently rapping, rapping at my chamber door.
'Tis some visitor,' I muttered, 'tapping at my chamber door' -

The use of alliteration can be seen in the first line: “nodded, nearly napping”. The use of the rhyming verbs “napping”, “tapping” and “rapping” have a powerful effect on the reader. Similar uses of repetition can be seen in songs. Repetition can be an effective lyrical tool. In his song “Handsome Johnny”, Richie Havens’ use of repetition emphasizes the repetitive nature of war. In each verse he uses the phrase

“Hey look there yonder, tell me what you see
marching to the Korean war”

In each verse, he simply replaces the name of the war with Gettysburg, Vietnam and others. He also repeats the phrase
“marching, still marching”

This line puts across the notion that war is unending and that soldiers continue to march through battlefields.

5. Truth

An interesting study by Schwartz [1982] investigated whether the truth value of a statement increases when it is repeated. Schwartz conducted an experiment during which she gave participants statements about people whose names were familiar and identifiable to them. All of the statements were of uncertain truth value, that is, that the validity of the sentence was neither clearly true nor false. Half of the statements were false and half of the statements were true. Participants were shown statements either once, twice or three times and later asked to rate the statements on a seven point truth value scale ranging from true to false. Schwartz found that participants were more likely to find a statement true the more it had been repeated. She also found that repetitions increased the participants likelihood to say that they had encountered the statement pre-experimentally [Schwartz, 1982].

This experiment and others like it have shown that by simply repeating a statement, a speaker can convince hearers of its validity. One setting where this can have serious repercussions is in the legal system. The results from these experiments suggest that a lawyer could sway a jury by simply repeating the statements which he wishes them to believe.

Smith and Meyer note the importance of repetition in evidence in trials. They say that jurors are often forbidden from note taking in trials and as such, repetition is of great importance in order that the jury remember and believe the arguments being put forward [Smith and Meyer, 1988].
2.4 The organisation of conversation

One of the fundamental concepts in conversation is that of turn-taking. Turn-taking organisation is a key component in an orderly conversation. Sacks, Schegloff and Jefferson [1974] constructed the most well-known models for how turn-taking is organised. In this study, they found that for the most part, one speaker talks at a time. Schegloff later expanded on this finding. He found that no matter how much overlap was found in a conversation, the overwhelming pattern was that one speaker spoke at a time. He also proposed the idea of “n-at-a-time”. He explained that there were activities for which it was intended that n people spoke at the same time. Two examples of this are chanting at sports games and congregational responses during religious services [Schegloff, 2000]. Sacks et al. found that turn-taking transitions are “finely coordinated”.

2.4.1 Turn Taking

Turn-taking is an important component in conversation. Turn-taking organisation describes how humans manage turn-taking in conversations. Turn constructional units (TCUs) and transition relevance places (TRPs) are the two major constituents of turn-taking.

- **Turn constructional units**

  According to Wooffitt, a turn constructional unit is a syntactical bounded lexical, clausal, phrasal or sentential unit [Wooffitt, 2005]. In addition, turn constructional units can also comprise of single words such as “huh?” or “hey!” as well as single phrases or clauses. Sacks et al. give some examples of such units:

  Fern: Well they’re not comin’,
  Lena: Who.
  Fern: Um Pam, unless they c’n find somebody.

  A property of turn constructional units is that speakers can generally anticipate when they are going to end [Wooffitt, 2005]. This allows speakers to
take up their turn without pauses between speaker utterances. Turn transfer takes place at transition relevance places.

• **Transition relevance places**

A transition relevance place occurs at the end of a turn constructional unit. It is at transition relevance places that speakers have the opportunity to take up the floor. How do speakers decide who should take up the floor? In their model of turn-taking organisation, Sacks et al. propose a set of techniques used in conversation in order to select the next speaker. These techniques are split into two categories:

**a)** *current speaker allocates the turn by selecting the next speaker*

Sara: Ben you want some?

Ben: Well alright I’ll have a, ((pause))

Sara: Bill do you want some?

Bill: No,

[Sacks et al., 1974]

Speaker selection is not always carried out using linguistic means. Gaze direction is an effective way for a speaker to allocate the next turn to another speaker.

**b)** *the next turn is self-allocated by self-selection*

Jim: Any a’ you guys read that story about Walter Mitty?

Ken: I did,

Roger: Mm hmm

[Sacks et al., 1974]

Sacks proposes a set of rules for the allocation of turns:

1. If a current-speaker-selects-next-speaker technique is used, then the party its use selects has rights to, and is obliged to, take next turn to speak, and all others are excluded.

2. If a current-speaker-selects-next-speaker technique is used, then on the next possible completion of the sentence the current speaker is con-
structing, transition should occur; i.e. current speaker should stop and next speaker should start.

3. If, by any next possible completion of the current sentence of a turn, current-speaker-selection of a next has not been done, self-selection may but need not be instituted, with the first starter acquiring rights to a turn at talk.

4. On any next possible completion of some current sentence, current speaker may stop, but unless he has done selection he need not stop unless another has self-selects.

The methods for allocating turns poses problems however. What happens when two (or more) speakers self-allocate the same turn? What happens when a transition relevance place is unclear or misleadingly projected by a speaker? In both of these cases we have overlapping speech.

2.4.2 Overlap

Overlap in conversation occurs for many different reasons. The three types of overlap examined below are simultaneous start-ups of turns by different speakers, back-channelling and interruptions. Jefferson [2004] states that although overlap appears to be messy and disorganised at first glance, it is, in fact, a very orderly feature of conversation.

- **Simultaneous start-ups**

Simultaneous start-ups occur when two (or more) speakers begin their utterances at the same time. This typically happens at possible transition relevance places, when speakers allocate turns by self-selecting. Speakers have two choices when simultaneous speech occurs: either they yield the floor, or they wait for the other speaker to yield the floor. How do speakers decide who should yield the floor? Schegloff suggests that it is of little importance to the organisation of the conversation who yields the floor, yet for the participants, it often is important. In order to hold the floor and essentially force the other speaker to stop speaking, speakers can employ
a number of techniques. Effective techniques are talking louder, talking at a higher pitch or changing the pace of the utterance in comparison to preceding utterances [Schegloff, 2000]. Although simultaneous start-ups are common, prolonged overlaps are rare. Generally, one speaker will yield the floor. However, who yields the floor typically depends on the participants of the conversation. Tannen [1994] speaks of “overlap-aversant” (those who stop speaking as soon as there is overlap) and “cooperative overlap” (those who overlap other speakers as a sign of cooperation) speakers.

- **Back-channelling**

Back-channels are short utterances by a speaker which indicate interest, attention and a willingness to listen [Gardner, 2001]. Back-channels are known by many other names: continuers [Schegloff, 2000], acknowledgment tokens [Jefferson, 2004] and response tokens [Gardner, 2001]. Drummond and Hopper suggest that the term “back-channel” does not sufficiently describe the many different types of these utterances.

“The failure . . . to distinguish between different classes of back-channels and the consequences they may have for speakership incipiency has made the back-channels category a hodgepodge – though the concept itself captures a basic intuition about brief turns. The concept remains widely cited, but evidence for its usefulness is thin and undifferentiated.”

(Drummond and Hopper quoted in Gardner [2001])

Gardner claims that there are at least seven categories of response tokens. These are:

- Continuers: *Mm hm, Uh huh*
- Acknowledgments: *Mm, Yeah*
- News markers: *Really?, Right*
- Change-of-activity tokens: *Okay, Alright*
- Assessments: *Great, How interesting*
– Brief questions: *Who?*, *Huh?*

– Non-verbal vocalizations and kinesic actions: *sighs, laughs, nods*

Gardner states that response tokens are not distributed evenly in conversation. They normally occur during extended utterances of speakers. Their frequency depends also on the type of conversation. They are much less frequently occurring in conversations in which turns change frequently (Sacks et al.'s “turn-by-turn” talk). Speakers also have their own styles of back-channeling. During the course of his research, Gardner found that some speakers produced up to 200 response tokens per hour, whereas others uttered less than 10 per hour [Gardner, 2001]. Gardner also found that response tokens frequently appear at transition relevance places.

• *Interruption*

Unlike other types of overlap, interruptions are seen as breaching the rules of turn-taking. That is, they are seen as a “violation of the current speaker’s right to be engaged in speaking” (West quoted in Bilmes [1997]). However, it is not universally agreed as to what constitutes an interruption as what one speaker feels is an interruption, another speaker may not. An interesting point made by Bilmes is that we often acknowledge the fact that we are violating the speaking rights of the other speaker. Utterances such as “can I just interrupt you there”, “I hate to interrupt but ...” and “sorry for interrupting” are common ways of “politely” interrupting another speaker. The following example by Hutchby (quoted in Bilmes [1997]) shows how an interruption takes place in natural conversation: (C is a caller to a radio talk show and H is the host)
C: I’m actually phoning in: support of the students, hh and also be
C: [curs I: -
H: [Wuh c-cacan
I just interrupt you, wu-
H: [w- were =
C: [yes = you actually on the demonstration yesterday?

How do speakers deal with being interrupted? From his research, Bilmes found that there are three main ways in which speakers display that their turn has been interrupted: direct claims, interruption displays and ignoring. Direct claims are where a speaker makes it explicit that they feel they have been interrupted. Utterances such as “can you just let me finish” or “I’m trying to tell you but you keep interrupting” make it clear to the interrupting party that the speaker feels that their speaking rights are being violated. These direct claims give weight to the argument noted above that interruptions are not clearly defined and different speakers find some utterances to be interruptions whereas others do not. Bilmes notes that if interruptions were clearly observable, we would not make direct claims that they have occurred. He points out that after being punched in the face, few people would say you hit me. An interruption is not as clear cut and, as such, we need to tell the interrupter that they have violated our right to speak.

Interruption displays are techniques employed by an interrupted speaker to show their annoyance at being interrupted or their determinism to continue
to hold the floor [Bilmes, 1997]. These displays can be both verbal and non-verbal. Examples of interruption displays are stopping mid-utterance or glares.

G: ... I'll make you a deal this evening. If you don't try to compare George Bush to Harry Truman, I won’t compare you to Jack Kennedy. ((laughter from audience and from Quayle))

G: Harry [Truman
Q: [you remember the last time someone compared themselves to Jack Kennedy? You remember what they said? (2)

G: Harry Truman is worth remembering . . . (from Bilmes [1997])

The above extract was taken from a debate between Quayle and Gore. When
Quayle interrupts him, Gore cuts off his own speech. He waits for Quayle to finish and then leaves a 2 second pause before taking up his turn. It is these displays that Bilmes says that Gore uses to show Quayle that he was annoyed by the interruption.

The third display technique employed by speakers to show that they have been interrupted is ignoring. Speakers can acknowledge that they have been interrupted by ignoring the interruption entirely and continuing to speak. Bilmes suggests that ignoring is the “sincerest form of insult” as to ignore somebody’s utterance, we must have acknowledged that it was uttered in the first place. We can only ignore something which we know has happened. The previous extract from the Gore-Quayle debate is an excellent example of ignoring an interruption. When Gore is interrupted by Quayle, he stops speaking and allows Quayle to finish his utterance. When Gore takes up his next turn, he continues the utterance that Quayle had interrupted, thereby ignoring the questions that Quayle asked him. He acts as if Quayle’s utterance never happened, even though it is clear that he heard the questions as he yielded the floor to him.

2.4.3 Gender and turn-taking

As mentioned previously, differences between speakers mean that frequency of overlap, interruptions or even the decision as to whether an utterance is in fact an interruption are very difficult to determine. One of the major difference between speakers’ styles of overlap is gender. Eckert et al. state that women tend to engage in cooperative, supportive conversation whereas males tend to engage in conversation in a competitive style. Males tend to talk in such a way that a hierarchy of conversational dominance is established, whereas women tended not to create hierarchies. [Eckert and McConnell-Ginet, 2003] It is perhaps unsurprising then that males and females engage in overlap in very different ways.

Mayer discusses a study of same-sex and cross-sex conversations by Zimmerman and West in which they noticed differences in the frequencies of overlaps between males and females. They found twenty-two overlaps and seven interruptions in the twenty same-sex conversations that they recorded and there were nine overlaps and
forty-eight interruptions in the eleven mixed-sex conversations. They noted that all of the overlaps and forty-six of the interruptions in the mixed-sex conversations were by males. Mayer says that this study has shown how males tend to ignore the rules of turn-taking when speaking to women, whereas woman are careful not to interrupt males. However, when speaking to other women, women tend to overlap.

Gender is thus an important factor to take into account in any study of overlap, as well as the whether the conversation is same-sex or cross-sex.
2.5 Film dialogue

The effects of repetition and overlap were not clear where film dialogue was concerned. As was shown in section 2, repetition and overlap do not appear to follow the pattern of natural conversation. Yet audiences do not report that the dialogue in films is odd or out of the ordinary. It seems that in suspending our belief, we also accept that the conversation that occurs in film fulfils a purpose. It was decided to analyse a brief extract from a film in order to prove whether or not film dialogue had the same patterns of repetition and overlap as natural conversation. The extract was chosen at random from the film “The Usual Suspect” and lasted for 7 minutes. There were five speakers, each of whom was given an identifier ‘n’, ‘k’, ‘y’, ‘d’ or ‘g’. These were the identifiers used during the main experiment. The experiment was carried out in the same manner as the main experiment which will be shown in section 3. The results of this experiment showed that reality had the effect of increasing the mean repetition (p <0.05), but that reality did not effect overlap significantly. However, it was found that in fact allo-repetition was not significantly affected by reality whereas self-repetition was affected significantly (p <0.001). This would indicate that like natural conversation, repetition is a feature of film dialogue, however it appears that only self-repetition is significantly affected by reality indicating that self-repetition is a feature of film dialogue whereas allo-repetition is not. Overlap does not appear to be a feature of film dialogue in the same way as natural conversation, as there was no significant difference between the shuffled transcripts and the original transcripts.

These results indicate that film dialogue behaves differently from natural dialogue. Yet, why do we not perceive this difference as odd? Why do audiences accept the conversation that takes place in films to be “real” conversation? This is perhaps an area which could be explored in the future.
2.6 Conclusion

Several phenomena of natural conversation have been examined above: turn-taking, overlap, repetition and priming. It has been shown that speakers tend to adhere to the rules which govern each of these phenomena. Speakers can also be seen to follow patterns regarding each of the phenomena, although differences between speakers exist. Speakers show frustration when other speakers violate the rules of conversation, as has been shown by the existence of utterances such as “you’re repeating yourself”, “don’t interrupt” and “I can’t get a word in edgeways”. The existence of priming has also been proved by many linguists using the lexical decision task and other such experiments. These phenomena are signs that conversation is not random or chaotic, but is in fact highly structured.
3 Main experiment

3.1 Introduction

The previous section described the many theories which aim to explain how speakers coordinate in conversation and provides explanations for the structure of conversation. However, one of the most common way of collecting this type of data is to run experiments in which participants are examined individually or grouped in pairs and then given a task. Although task-driven experiments provide much useful data, the participants are somewhat limited in the vocabulary they can use and the flow of conversation is different to that of natural conversation as they are talking for the sole purpose of completing a task and only speak on that topic.

This experiment is different from all others which were researched during the course of this project in that the data analysed was natural conversation and the participants were in a group setting. The participants were given no task other than to talk. They had the freedom to choose the topic and to change it as they pleased.

In this section, the experiment itself will be described, as will the corpus used during the analysis including the participants and the content of the conversations recorded.

3.2 Aims

The first aim of this experiment was to analyse the structure of natural conversation. The theories which attempt to explain this structure only analyse conversation in its natural order. In this experiment, natural conversation was compared with the same conversation in a shuffled order to establish whether the phenomena such as priming, repetition and overlap were only found in natural conversation or if in fact they were also present in a shuffled conversation.

The second aim of this experiment was to create a program which could analyse these phenomena. Although it is quite time-consuming, there is considerable freedom and flexibility associated with the creation of one’s own analysis program. Also, as this experiment is novel, there is no such existing software which could
measure these phenomena.

3.3 Transcripts

3.3.1 Choosing a corpus

The choice of corpus was important. In order to choose a corpus to work with, one must consider the purpose of the analysis. A linguist might need to consider the content of a corpus, as they are generally interested in the nature of the material in the corpus.

A computational linguist on the other hand, is not concerned about the exact nature of the material. As this project was not interested in interpreting what speakers said, a computational linguistic approach was taken. This meant that the choice of corpus was less restricted. Many corpora were examined including the two described below:

- **ICE-Ireland**
  ICE-Ireland is the Irish corpus which forms part of ICE (International Corpus of English). ICE is a large collection of English corpora. Teams around the world are compiling regional or national corpora of English to add to ICE. A team from Trinity College Dublin and Queen’s University Belfast have recently created ICE-Ireland, a corpus of spoken and written Hiberno-English. The corpus contains 1 millions words which are contained within 500 texts, 300 of which are spoken texts and 200 of which are written texts.¹

- **MICASE**
  The Michigan Corpus of Academic Spoken English or MICASE is a corpus of 200 hours of recordings (1.8 million words). The recordings were taken in the University of Michigan and are all of academic speech events, such as lectures, seminars and advising sessions. The corpus allows users to choose transcripts which appeal to their needs. Users can browse the corpus by academic position/role, native speaker status, first language, speech event

type, academic division, academic discipline, participant level and interactivity rating. ²

Neither ICE-Ireland nor MICASE suited for the purpose of this project. Although they both contain vast amounts of data, neither corpus contained transcripts in which the timings of the utterances were noted. As this project concerns the phenomena of overlap, the times markings of utterances were vital.

3.3.2 Corpus used

The corpus used for this project was kindly provided by Professor Nick Campbell.³ The corpus was entitled “table talk” and contained the recordings of three days of natural conversation. Each of these conversations was transcribed by a member of Professor Campbell’s team and was marked for time.

The conversation on day 1 took place between four speakers and lasted for 34 minutes and 33 seconds. On day 2, the conversation took place between five speakers and lasted for 1 hour 22 minutes and 14 seconds. On day 3, the conversation took place between the four speakers from day 1 and lasted for 1 hour 22 minutes and 42 seconds. Each speaker was given an identifier. Nick was ’n’, Damien was ’d’, Christina was ’k’, Izumi was ’y’ and Lorene was ’g’. Nick, Damien, Christina and Izumi were present for all three days, while Lorene was only present on day 2. The speakers will be discussed in section 3.4 below.

Each line of the transcripts contained the start and end time of the utterance, the speaker of the utterance, information about the emotions of the speaker and the utterance itself.

00:57:20 00:57:23 n 3 3 Inside the house? Right, yeah.
(day 2-line 1524)

The transcript was ordered by the start times of the utterances, then by the end times and then by speaker if necessary.

²http://micase.elicorpora.info/
³This data was supplied by Professor Nick Campbell (personal communication; nick@tcd.ie); www.speech-data.jp – last verified: April 2010.
The speakers spoke in English for the majority of the conversation, although many Japanese words were used throughout the conversation. The participants discuss the meanings of some Japanese terms such as “oubeika”. All Japanese words are written in *romaji* which is the romanization of Japanese using the Latin alphabet. It is used for foreigners learning Japanese in order to aid them in the pronunciation of Japanese words before they learn how to read the three alphabets used in Japanese (katakana, hiragana and kanji).

### 3.3.3 Changes made to the transcripts

The transcriptions were relatively well ordered although some changes had to be made to them in order to make them easily readable by the analysis program. The format of start and end times had to be changed in each of the three transcripts as they were not consistent throughout. Each time was changed to the format hr:min:sec. This meant that the time 26:35 was changed to 00:26:35. Some characters also had to be changed as they could not be read by the program, such as (...) in place of the single symbol (...).

Spell checking also had to be carried out as the transcriber of the conversations was not a native English speaker. These changes included grammatical mistakes such as replacing “there’re” with “they’re”, and general spelling errors. Ensuring that the transcripts were uniform was of high importance as the results of the analysis would be affected if the program read the transcripts incorrectly due to anomalies.

### 3.4 Participants

The participants of the conversations were Nick (Professor Campbell), Damien, Christina, Izumi and Lorene. Nick, Damien, Christina and Izumi took part in the conversations of days 1, 2 and 3, while Lorene only participated in the conversation of day 2. Nick and Damien worked in the language laboratory in which the conversations were recorded. Christina is a linguist but does not work in the laboratory. Izumi was a guest and Lorene was a friend of Izumi’s who she invited to join the team for day 2. Neither Izumi nor Lorene were familiar with the setting.
of a linguistic laboratory. The participants are all of different nationalities. Nick is English, Damien is Belgian, Christina is Finnish, Izumi is Japanese and Lorene is Australian.

3.5 Conversations

The conversations which took place over the three days were informal and unplanned. The only occurrence of conversation planning happened at the end of day 1. Izumi asked Nick if she could bring her friend along with her the next day. Nick agrees but Izumi is concerned about conversation topics as her friend did not know very much about Japanese culture. The other participants suggest that they can talk about topics other than the ones they had talked about during the first day’s conversation. Although topics were suggested, the participants can lead the conversation in any direction that they wish.

00:33:13 00:33:18 y 3 4 hum, but the problem is she doesn’t understand about Manga
00:33:18 00:33:19 y 3 3 or I ...
00:33:19 00:33:20 d 4 4 we can talk about something
00:33:19 00:33:20 n 3 4 does she speak English?
00:33:19 00:33:22 k 4 4 no, but we, to, hum, ...different things tomorrow, yeah
(day 1 lines 1300-1304)

The conversation topics on day 1 range from the meaning of “Oubeika” and “manzai” to the different television stations that the participants have seen while traveling the world. The conversation mainly relates to cultural differences, especially Japanese cultural differences. On day 2, the conversation takes a different direction than day 1. The overall conversation is about technology, including talk about solar panels and Google. The conversation diverges at times to cultural differences. On day 3, the participants talk mainly about Japanese life and language. They also talk about Japan from the point of view of a visitor. The conversations were recorded in a laboratory in Japan and as such, the topic of conversation is mainly related to Japan or Japanese. The participants are all of different nation-
alities and they spend long periods of time talking about cultural differences.

3.6 Programming language used

The language chosen for the writing of the program was C. C is one of the most popular programming languages in use \(^4\) and has influenced many of the other most popular languages such as C++, Java and Python. \(^5\) C allows access to low-level memory, yet also provides high-level control of data structures and procedures [Banahan, 2003]. The vast majority of new operating systems are written in C (and/or C++) and C is hardware independent, that is, it is portable to most operating systems [Deitel and Deitel, 2010]. Although Perl is often the first choice programming language for programmers analysing linguistic data, the flexibility of C was the deciding factor in choosing a programming language to write the experiment’s analysis program in.

3.7 Program design

The program began by splitting up the text line by line. Each line was tokenised and entered into a data structure. Each structure contained a start time, end time, speaker identifier, emotion 1, emotion 2 and the utterance spoken. Each utterance was then tokenised into its separate words. An array was created to hold each of these words. The number of words in each utterance was counted and this number was also placed in the struct for each line. Contractions such as “i’m” and “couldn’t” were counted as one word, not two. An overall array called dataArray containing a structure for each line was then created.

3.7.1 Shuffled Transcripts

Shuffled versions of each of the three transcripts were created within the program. These transcripts were created using the “bags” approach. The “bags” approach is based on the idea of having a bag full of distinct numbers. A number is chosen from the bag and is not replaced. This continues until the bag is empty.

\(^4\)http://www.langpop.com/
\(^5\)http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html
In order to carry out the shuffling of the transcripts using this approach, the number of lines must first be known. This number was calculated and all numbers between 1 and the number were placed in a linked list. Next, the C random number generator was used to generate a number between 0 and the number of lines in the file. The node at the position indicated by the randomly generated number was selected from the linked list. The value from this node was placed into a new array. The node was then deleted from the linked list to ensure that this number was not selected again. The pool of numbers that the random number generator could choose from was decremented by 1. This continued until all of the numbers in the “bag” had been selected.

A new array containing the numbers 1 to the number of lines in the file in a random order had now been created. However, the random number generator in C (and in many other programming languages) is pseudo-random. Computers cannot generate truly random numbers [Gentle, 2003]. In order to generate a “random” number, it uses an mathematical formula. This means that the numbers generated are not truly random, but are generated in a predictable fashion. The solution to this problem was to create 10 shuffled transcripts for each real transcript and find the average of the analysis for the shuffled transcripts.

### 3.7.2 N-gram data structures

N-grams are data structures composed of sequences of \( n \) number of items from a given sequences. N-grams containing 1 item are called unigrams, 2 items are called bigrams, 3 items are called trigrams and 4 or more items are simply called n-grams. The exact nature of an item is decided by the programmer. Items can be single characters, words or sentences. In the case of this program, an item was a word. Kowalski and Maybury show how the words “sea colony” can be searched using bigrams and trigrams:

- bigrams: se ea co ol lo on ny
- trigrams: sea eac aco col olo lon ony

[Kowalski and Maybury, 2000]
One of the uses of n-grams is in spelling error detection. The most common approach to this process is by using statistics to flag any word which contains n-grams (typically trigrams) which are particularly rare in the language being looked at [Kowalski and Maybury, 2000]. These words might contain spelling errors.

The use for n-grams in this project was to be able to separate repetitions such as “the” and “a” from longer repetitions. The value of using n-grams will be examined later in section 5

### 3.7.3 Analysis

The program analysed the transcript two lines at a time, comparing them to each other, allowing each line to be compared with the preceding and following lines. Firstly, the program checked if there was any repetition between the two lines. A repetition was described as being an identical token. Each token was a n-gram. The analysis was repeated for n = 1 up to n = 5. This means that sequences of one, two, three, four and five words were analysed. If any repetition was found, the program noted whether it was a case of allo-repetition or self-repetition.

**1-gram self-repetition:**
- 00:04:41 00:04:44 d 3 3 but Mukashika, if you would say Ou bei ka
- 00:04:44 00:04:45 d 3 3 you have...
(days 1-lines 164/165)

**2-gram allo-repetition:**
- 00:14:20 00:14:22 y 3 3 and come back for volume three
- 00:14:21 00:14:21 n 3 3 volume three
(days 3-lines 529/530)

**5-gram allo-repetition:**
- 01:11:03 01:11:09 d 3 3 just land in a residential area and could be the size of this room
- 01:11:05 01:11:09 n 3 3 About the size of this room basically, but...x.
(days 2-lines 1792/1793)
Next, the program checked for any instances of overlapping speech in the transcripts. Once again, the transcripts were compared two lines at a time. The start and end times of both lines were compared to see if they overlapped. An overlap was described as:

- **equal start times and end times**
  00:24:33 00:24:34 k 3 2 there is a studio
  00:24:33 00:24:34 d 3 2 it’s too heavy?
  (day 1 lines 928/929)

- **equal start times or end times**
  00:56:45 00:56:47 g 3 3 That was 5 years ago, no?
  00:56:45 00:56:48 y 3 3 Yes, uh...um.
  (day 2 lines 1513/1514)

- **the start time and/or end time of one line is between the start and end time of the other line**
  00:02:41 00:02:43 y 3 3 but now she needs one for herself @w
  00:02:42 00:02:44 n 3 4 for herself @w
  (day 3 lines 96/97)
  00:31:39 00:31:42 d 3 3 when I look at students they still buy quite lot of Manga
  00:31:40 00:31:41 n 3 3 you working on time?
  (day 1 lines 1217/1218)

- **the end time of one line is between the start and end time of the other line**
  00:44:02 00:44:03 n 3 3 @w
  00:44:02 00:44:10 y 3 4 sesame oil from on the top of the hum negi ..onion spring onions
  (day 3 lines 1401 and 1402)

The length of each overlap was calculated and recorded, and the overlap counter was incremented. The total number and time of overlaps per speaker was also calculated.
During the analysis of the randomised transcripts, a problem was noticed. The counter for overlap was almost always zero. This was due to the unlikeliness of two lines from the original transcript being next to each other in the randomised transcripts. The solution to this problem was to check for “fake overlap”.

Fake overlap was the overlap that occurred when only the seconds of the start and end times were taken into account. Using the original definitions of overlap, the following lines would not be found to be overlapping:

00:56:45 00:56:47 g 3 3 That was 5 years ago, no?
01:18:46 01:18:52 d 3 3 then you will have a sort of light well, to get some lights in the lower levels.

However, when only the seconds were taken into account, these lines became:

45 47 g 3 3 That was 5 years ago, no?
46 52 d 3 3 then you will have a sort of light well, to get some lights in the lower levels.

When only the seconds were taken into account, an overlap was found. The original transcripts were also analysed in this way so that real overlap and fake overlap could be compared between the real and shuffled transcripts. The results of real overlap and fake overlap were slightly different in the real transcripts due to cases such as the following:

01:04:59 01:05:02 y 3 3 Yes, sometimes...not, not...x
01:05:01 01:05:04 n 3 2 x...@w

When analysing real overlap, these lines would be found to be overlapping. However, using the fake overlap calculation, they would not be found to be overlapping:

59 02 y 3 3 Yes, sometimes...not, not...x
This is due to the way that the program calculated overlap. In the real overlap calculation, the second line’s start time would have been found to be between the first line’s start and end times. However, in the fake overlap calculation, the program found that neither the start time nor the end time of the second line occurred during the start and end times of the first line. These types of cases were the only cases where real overlap was found but fake overlap was not.

When an overlap was found, the program checked whether there was repetition by speakers during the overlapping utterances. This showed that speakers were repeating the same word (or words depending on the value of n) as another speaker at the same time.

A file containing all of the results was created. Each line of this file gave the results for the analysis of two lines in a transcript. There were 15 columns in the file:

1. day—the day being analysed
2. i—the line number of the first line in the pair being analysed
3. n—the value of n at the time
4. curr—the speaker at line i
5. next—the speaker at line i+1
6. self—a binary value which indicates if the repetition is self-repetition
7. allo—a binary value which indicates if the repetition is allo-repetition
8. rep—a binary value which indicates whether or not there is repetition

9. sharedtokens—the number of tokens that the two lines have in common

10. totaltokens—the total number of tokens between the two lines

11. reality—the original transcript was given the value 0, while the 10 shuffled transcripts had values of 1 to 10

12. overlap—a binary value which indicates if there was overlap between the two lines

13. repoverlap—a binary value which indicates if there was repeated overlap between the two lines

14. fakeoverlapcheck—a binary value which indicates whether or not there is fake overlap between the two lines

15. fakerepoverlap—a binary value which indicates whether there is fake repeated overlap between the two lines

3.8 Calculating results

During the analysis of each pair of lines from the transcripts, the 15 variables mentioned above were changed according to the analysis. For example, if there is self-repetition (the speaker at i is the same as the speaker at i+1) between two lines, the variable self is given the value of 1. Similarly, if there is overlap between the two lines, the variable overlap is given the value of 1. When a pair of lines were analysed, a new line was added to the results file. This line contained the values for all 15 variables that were analysed. The results file when completed contained almost 330,000 lines of data. This file had to be analysed automatically using software called R. R is a high-level programming language and also an environment for statistical analysis [Crawley, 2007]. R was designed with the influence of two existing languages: S and Scheme.

The variables day and n were treated as categorical data rather than ordinal data by using the factor( ) method of R. Reality was treated as a binary value.
The value 0 was assigned TRUE while the values 1-10 were assigned the value FALSE. Binary response variables such as \textit{rep} and \textit{self} were then examined with respect to these factors. A generalised linear model was fitted using one binary response variable and these factors. The control set was a combination of the relevant factors from those that follow: reality = false, day 1, n = 1 and speaker = d.

3.9 Conclusion

The aims of this experiment were to analyse the structure of natural conversation by measuring the frequency of the phenomena of priming, repetition and overlap which were described in section 2. Previous experiments which analysed these phenomena only concentrated on conversation in its natural order. This experiment aimed to compare conversation in its natural order to conversation in a shuffled order. The purpose of this comparison was to investigate whether these phenomena were only a feature of natural conversation as the theories in section 2 suggests, or if in fact there were also found in the unordered versions of the conversations.

Although the creation of the program was time-consuming, the implementation was extremely successful as all of the aims of the experiment were met.
4 Results

4.1 Introduction

The results obtained by the analysis program are discussed in this section. The results for repetition are shown first. These results are displayed by day and then an average set of results is given. The results for overlap are given after the repetition results. The first overlap result given is the overlap and fake overlap by day for the original and shuffled transcripts. The results for repeated overlap are then given by day, followed by the average results for repeated overlap. All of the statistical results were calculated using reality = false, day 1, n = 1 and speaker = d as the control set.

4.2 Repetition

The results for the repetition analysis are depicted below in figures 1, 2, 3 and 4. The x-axis of each graph represents the value of n and the y-axis represents the number of occurrences of repetition. Four variables were measured: repetition in the original transcript, self-repetition in the original transcript, repetition in the shuffled transcript and self-repetition in the shuffled transcript. The results for each day are shown below, as is a graph indicating the average values of repetition over the three days. The results for the shuffled transcript is the average result calculated from the analysis of 10 shuffled transcripts, the reasons for which are stated in section 3.7.1.

4.2.1 Day 1

Figure 1 shows the occurrences of the repetition variables for day 1 for increasing values of n. For n = 1, there are 662 cases of repetition between lines of the original transcript, 250 of those being self-repetitions. In the shuffled transcript, we find 361 cases of repetition, 137 of those being self-repetitions.

When n was increased to 2, the values for repetition in the original transcript decreased to 97 while the value for allo-repetition in the shuffled transcript decreased to 57. The values for self-repetition in both transcripts also decreased.
The values for self-repetition was 13 in the original transcript and 10 for the shuffled transcript.

N was then increased to 3, and a change in values was seen once again. The number of occurrences of repetition in the original transcript was 27, 17 of which were self-repetitions. The shuffled transcript contained 6 repetitions, 3 of those being self-repetitions.

When the value of n was increased to 4, the number of repetitions in the original transcript dropped to 10, all of which were self-repetitions. There were no occurrences of repetition in the shuffled transcript.

Finally, n was increased to 5. 9 repetitions were found in the original transcript, all of which were self-repetitions. Again, no occurrences of repetition were found in the shuffled transcript.

Looking at the data in more detail we find that for all values of n, the number of repetitions in the original transcript is greater than that of the shuffled transcript. There are 1.83 times more repetitions in the original transcript when n was 1, 7.46 times more when n was 2 and 4.5 times more when n was 3. There were no repetitions in the shuffled transcript when n was 4 or 5.

The percentage of self-repetitions was also calculated. In the original tran-
script, the percentage of repetitions being self-repetitions was 37.76% when n was 1, 54.64% when n was 2, 62.96% when n was 3 and 100% when n was 4 and 5. In the shuffled transcript, the percentage of repetitions being self-repetitions was 37.95% when n was 1, 76.92% when n was 2 and 50% when n was 3.

4.2.2 Day 2

![Graph](image)

Figure 2: Occurrences of repetition on day 2 for n = 1 to n = 5

Figure 2 plots the results of the repetition analysis on day 2 for the five values of n being analysed. When n was 1, there were a total of 1050 occurrences of repetition in the original transcript, 225 of those being self-repetitions. 803 occurrences of repetition were found in the shuffled transcript of which 219 were self-repetitions.

N was then increased to 2 and 119 cases of repetition were found in the original transcript. Of these, 22 were self-repetitions. 35 cases of repetition were found in the shuffled transcript and 11 of these were self-repetitions.

N was increased once again, to a value of 3. 44 repetitions were found in the original transcript, 9 of which were self-repetition. 7 repetitions were found in the shuffled transcript and 3 of these were self-repetitions.
N was then increased to 4. 21 occurrences of repetition were found in the original transcript, 6 of which were self-repetitions. No occurrences of repetition were found in the shuffled transcript.

For n = 5, 11 cases of repetition were discovered. 5 of these repetitions were self-repetitions. As with n = 4, no cases of repetition were found in the shuffled transcript.

For all values of n, the number of repetitions was greater in the original transcript than in the shuffled transcript. When n was 1, there were 1.31 times more repetitions in the original transcript. When n was 2, the number of repetitions in the original transcript was 3.4 times greater than the shuffled transcript. When n was 3, there were 6.29 times more repetitions in the original transcript than in the shuffled transcript.

As with day 1, the percentage of self-repetitions was calculated. When n was 1, 21.43% of the total number of repetitions were self-repetitions in the original transcript. For n = 2, the percentage was 18.45%, 20.45% for n = 3, 28.57% for n = 4 and 45.45% for n = 5. In the shuffled transcripts, the percentage of repetitions being self-repetitions was 27.27% when n was 1, 31.43% when n was 2 and 42.86% when n was 3.

4.2.3 Day 3

The final day was then analysed for repetition and these results are shown in figure 3. When n was 1, 1210 cases of repetition were detected. 567 of these repetitions were self-repetitions. In the shuffled transcript, 679 repetitions were detected and 297 were self-repetitions.

N was then increased to 2 and 178 cases of repetition were observed, 90 of which were self-repetitions. In the shuffled transcript, 37 occurrences were observed and 17 of these were self-repetitions.

For n = 3, 43 repetitions were found in the original transcript. 25 of these were self-repetitions. 12 repetitions were found in the shuffled transcript and 8 of these were self-repetitions.

For n = 4, 15 repetitions were found in the original transcript and 9 of these were self-repetitions. No occurrences of repetitions were found in the shuffled
Finally, when \( n \) was increased to 5, 6 cases of repetition were found in the original transcript. 2 of these repetitions were self-repetitions. Once again, no repetitions were found in the shuffled transcript.

Once again, it can be seen that the number of repetitions was greater in the original transcript than in the shuffled transcript. There are 1.78 times more repetitions when \( n \) was 1, 4.81 times more repetitions when \( n \) was 2 and 3.58 times more repetitions when \( n \) was 3.

The percentage of self-repetitions in the transcripts was calculated for day 3. Self-repetitions accounted for 46.89% of repetitions in the original transcript when \( n \) was 1, 50.56% of repetitions when \( n \) was 2, 58.14% of repetitions when \( n \) was 3, 60% of repetitions when \( n \) was 4 and 33% of repetitions when \( n \) was 5. In the shuffled transcript, self-repetitions were calculated to be 41.09% of repetitions when \( n \) was 1, 45.95% of repetitions when \( n \) was 2 and 66.67% of repetitions when \( n \) was 3. There were no repetitions in the shuffled transcript when \( n \) was 4 or 5.
4.2.4 Average repetition

The overall results from the three days of conversations were also calculated and have been shown in figure 4. The results for the original transcripts are the average of the results from days 1, 2 and 3. The results for the shuffled transcripts were calculated in an identical fashion, however, it should be remembered that the results for each day were the average of 10 shuffled transcripts. This means that the average results for all three day’s shuffled transcripts were in fact the average results for 30 shuffled transcripts, 10 from each day of conversation.

When n was 1, the average number of repetitions in the original transcripts was 974, of which 347 were self-repetitions. The number of repetitions in the shuffled transcripts was 614, of which 218 were self-repetitions.

131 repetitions were found in the original transcripts when n was 2. 55 of these were self-repetitions. In the shuffled transcripts, 28 occurrences of repetition were found with 13 being self-repetitions.

N was further increased to 3. 38 repetitions on average were found in the original transcripts. On average, 17 of these were self-repetitions. In the shuffled
transcripts, 8 repetitions were found with 5 being self-repetitions.

When n was 4, 15 repetitions were found in the original transcripts. Of these repetitions, 8 were self-repetitions. There were no occurrences of repetition in the shuffled transcripts when n was 4.

Finally, n was increased to 5 and 9 occurrences of repetition were found in the original transcript. 5 were self-repetitions. There were no repetitions in the shuffled transcripts when n was 5.

The average repetition results showed clearly that there were more repetitions in the original transcripts than in the shuffled transcripts. When n was 1, there were 1.59 times more repetitions in the original transcript. There were 4.68 times more repetitions when n was 2 and 4.75 times more when n was 3. There were no cases of repetition in the shuffled transcripts when n was 4 or 5, whereas there were 15 cases when n was 4 and 9 cases when n was 5 in the original transcripts.

When calculating the percentage of self-repetitions in the original transcripts,
a pattern emerged. The percentage of self-repetitions increased as \( n \) increased. When \( n \) was 1, 35.63% of the repetitions were self-repetitions. This figure increased to 41.98% when \( n \) was 2, to 44.74% when \( n \) was 3, to 53.33% when \( n \) was 4 and to 55.55% when \( n \) was 5. A similar pattern was seen in the shuffled transcripts. 35.5% of repetitions were self-repetitions when \( n \) was 1, 46.34% when \( n \) was 2 and 62.5% when \( n \) was 3. As stated previously, there were no repetitions when \( n \) was 4 or 5.

The results of repetition were examined using the statistical program R (see Section 3.8). Each of the factors day, \( n \) and reality were examined in isolation. It was found that reality had a significant effect on repetition, increasing the mean \((p < 0.01)\). The value of \( n \) also proved to be significant in some cases. When \( n \) was 2, 3 and 4, there was a significant effect on repetition, with all values decreasing the mean repetition \((p < 0.001 \text{ in all cases})\). Day was also found to be significant, with day 2 increasing the mean \((p < 0.001)\) and day 3 decreasing the mean \((p < 0.05)\).

Next, the interaction between factors was examined. The interaction between reality and \( n \) was found to be significant when \( n \) was 2 and 3, with both values of \( n \) increasing the mean repetition when interacting with reality \((p < 0.001 \text{ in both cases})\). Day and \( n \) also interacted to have a significant effect on repetition, increasing the mean when \( n \) was 3 on day 2 \((p < 0.001)\), increasing the mean when \( n \) was 3 on day 3 \((p < 0.001)\) and decreasing the mean when \( n \) was 4 on day 3 \((p < 0.001)\).

The interaction of reality, day and \( n \) was also examined. On day 3, when \( n \) was 3 and reality was true, there was a significant decrease of the mean repetition \((p < 0.001)\). Also, on day 3, when \( n \) was 4 and reality was true, the mean repetition was significantly affected, increasing for this interaction \((p < 0.001)\). The effects of these interactions can be seen in figure 5.

The results from the analysis of overlap in the transcripts are presented below. Figure 6 shows the occurrences of overlap over each of the three days of conversations. The x-axis indicates the particular day being analysed and the y-axis indicates the number of occurrences of overlap. The variables that were analysed were “real” overlap and “fake” overlap in both the original and shuffled transcripts.

Figures 8, 9 and 10 show the number of occurrences of repeated overlap in
days 1, 2 and 3 respectively. Figure 11 is a graph which shows the average number of repeated overlaps calculated over the three days of conversation. Repeated overlap is calculated using “fake” overlap. The x-axis on each of these four graphs shows the value of n. The y-axis shows the number of repeated overlaps.

4.2.5 Real and fake overlap

![Graph showing occurrences of overlap by day](image)

In figure 6, the number of “real” and “fake” overlaps in the original and shuffled transcripts are shown. On day 1, there were 319 occurrences of “real” overlap and 315 occurrences of “fake” overlap in the original transcript. On day 2 in the original transcript there were 581 cases of “real” overlap and 569 cases of “fake” overlap. Finally, on day 3 in the original transcript there were 415 cases of “real” overlap and 400 cases of “fake” overlap.

In the shuffled transcript for day 1, there were 0 cases of “real” overlap and 31 cases of “fake” overlap. On day 2, there were 0 cases of “real” overlap and 76 cases of “fake” overlap. On day 3, there were 0 cases of “real” overlap and 47 cases of “fake” overlap.
As can be seen clearly from the graph, there are more “real” and “fake” overlaps in the original transcripts than in the shuffled transcripts.

The results of ‘real” overlap were then examined using R. The factors day and reality were examined in isolation. It was found that reality had a significant effect on “real” overlap, increasing the mean (p < 0.001), as did day: (p <0.001 on days 2 and 3, with day 2 increasing the mean and day 3 decreasing the mean).

Next, the interaction between these factors was examined. On day 3, reality interacted to significantly reduce the mean “real” overlap (p <0.001).

Next, the results for “fake” overlap were examined. As with “real” overlap, reality was found to have a significant effect on “fake” overlap, increasing the mean (p <0.001). Day 2 and day 3 also played a significant role in “fake” overlap. The effect of day 2 and day 3 was to increase the mean (day 2: p <0.001 and day 3: p <0.1). Reality and day interacted once again to have a significant effect on overlap. On day 2, when reality was true it was found that the mean was increased...
significantly ($p < 0.001$). On day 3 when reality was true, it was found that the mean decreased significantly ($p < 0.001$). The effects of these interactions can be seen in figure 7. The graph also shows that the speaker has an effect on “fake” overlap, however, these effects have not yet been examined. Section 2.4.3 noted, there are differences in the overlapping patterns between males and females. However, some literature noted that different nationalities have different overlapping patterns, showing that there are a number of possible reasons for the different effects of speakers on overlap. A future continuation of this experiment could examine the effect of the speaker on overlap.

4.2.6 Day 1

Figure 8: The occurrences of repeated overlap for day 1

Figure 8 shows the number of overlaps with repetition in the original and shuffled transcripts for day 1. In the original transcript, there were 97 cases of repeated overlap when $n$ was 1. There were 11 cases of repeated overlap when $n$ was 2, 4 cases when $n$ was 3, 1 case when $n$ was 4 and 1 case when $n$ was 5.

In the shuffled transcript, there were 13 cases of repeated overlap when $n$ was 1, 1 case when $n$ was 2 and no cases when $n$ was 3, 4 or 5.
4.2.7 Day 2

Figure 9: The occurrences of repeated overlap on day 2

On day 2, as illustrated in figure 9, there were 207 cases of repeated overlap in the original transcript when \( n \) was 1, 23 cases when \( n \) was 2, 12 cases when \( n \) was 3, 5 cases when \( n \) was 4 and 2 cases when \( n \) was 5.

In the shuffled transcript, there were 25 cases of repeated overlap when \( n \) was 1, 1 case when \( n \) was 2 and no cases of repeated overlap when \( n \) was 3, 4 or 5.

4.2.8 Day 3

Figure 10 shows the repeated overlap occurrences in the original and shuffled transcripts from day 3. In the original transcript, there were 130 cases of repeated overlap when \( n \) was 1, 19 cases when \( n \) was 2, 4 cases when \( n \) was 3 and no cases of repeated overlap when \( n \) was 4 or 5.

In the shuffled transcript, there were 16 cases of repeated overlap when \( n \) was 1, 1 case when \( n \) was 2 and no cases of repeated overlap when \( n \) was 3, 4 or 5.
4.2.9 Average overlap

Figure 11 shows the average number of repeated overlap in the original and shuffled transcripts over the three days of conversation. As with the analysis of repetition, it should be noted that as there are 10 shuffled transcripts for each original transcript, the average result for the shuffled transcript is the average of 30 shuffled transcripts.

In the original transcripts, the average number of repeated overlaps when \( n \) had the value of 1 was 145. When \( n \) was increased to 2, the number of repeated overlaps dropped to 18. When \( n \) was 3, there were 7 cases of repeated overlap found by the program. There were 2 cases when \( n \) was 4 and 1 case when \( n \) was 5.

In the shuffled transcripts, there were on average 18 repeated overlaps when \( n \) was 1. When \( n \) was 2, there was 1 case of repeated overlap. On average, there were 0 cases of repeated overlap when \( n \) was 3, 4 and 5. The statistical analysis of the results for repeated overlap included an analysis of both “real” and “fake” repeated overlap. It was found that reality had a significant effect on “real” and “fake” repeated overlap, increasing the mean in both cases (\( p < 0.001 \) and \( p < 0.01 \) respectively). Day 2 had a significant effect on “real” repeated overlap, increasing the mean (\( p < 0.001 \)) and on “fake” repeated overlap, increasing the mean also (\( p < 0.001 \)).
Figure 11: The average number of occurrences of repeated overlap

<0.001). Day 3 also had a significant effect on “real” and “fake” repeated overlap, decreasing the mean in both cases (p <0.05 and p <0.001 respectively). The effects of these factors can be seen in figure 12.
Figure 12: The effects of reality, n and day on fake repeated overlap
5 Discussion and Analysis of results

5.1 Introduction

The previous chapter reported the results as calculated by the analysis program. However, it is necessary to understand what these results signify and as such, this chapter will discuss these results in the context of the theories and research which were presented in Section 2. The results for repetition will be discussed firstly, followed by the results of overlap and then the results for repeated overlap.

5.2 Discussion and Analysis

5.2.1 Repetition

Two hypotheses were presented in relation to repetition:

- Greater numbers of repetitions would be seen in the original transcripts than in the shuffled transcripts.

- Less repetition would be found as n increased

In Section 4, we saw that there were more cases of repetition in the original transcript than the shuffled transcript for all values of n. In fact, on day 1 when n was 2, there was almost 8 times more repetition in the original transcript than in the shuffled transcript. On day 2, when n was 3 there were over 6 times more repetitions in the original transcript than in the shuffled. On day 3, there were almost 5 times more repetitions of trigrams in the original transcript than in the shuffled transcript. The smallest difference between the number of repetitions in the original transcripts and the shuffled transcripts came on day 3. When n was 1, there were 1.31 times more repetitions in the original transcript. Given that the next smallest difference was 1.78 times (day 3  n = 1), the repeated words from day 2’s shuffled transcripts were briefly examined.

It was found that there was a large number of “x”s throughout the transcript for day 2. An “x” means that the transcriber could not fully hear the utterance. In the analysis of the shuffled transcript, there were on average 192 cases per transcript of “x” being repeated. To put this into perspective, on day 3 there
were on average 4 cases per shuffled transcript and on day 1, there were no cases within the 10 shuffled transcripts of “x” being repeated. If the average number of repeated “x”s was taken to be 2 (the average of repeated “x”s between days 1 and 3), the value of repetitions in the shuffled transcript for day 2 when n was 1 would have been 613. This would have meant that there were 1.72 times more repetitions in the original transcripts than in the shuffled transcripts. This result would have been more in line with the other two days of conversation. This highlights how something as seemingly simple as the notations used by transcribers can have a marked effect on the results.

Even when the results which include the repetition of “x” are used, we can see that for each day, and for all values of n, that the number of repetitions in the original transcripts is greater than that of the shuffled transcripts. This result was also found to be significant during analysis of the repetition results using R.

In Section 2.2, we explored the phenomenon of priming. It was shown that priming meant that words, phrases and syntactic structures of one utterance could influence the interpretation or production of another. In Section 2.3, it was shown that repetition had many functions in conversation, one of which was confirmation and clarification. Speakers repeated utterances of other speakers in order to signal to the other speaker that they had not fully understood or heard what they had said. Linking priming with repetition, in such a situation we would be more likely to find the pair of utterances in (a) than (b) (see section 2.2.2).

1. A: Can you pass me the book please?
   B: The book?

2. A: Can you pass me the book please?
   B: A book?

N-gram analysis was explored as it was thought that the 5-gram repetitions would be much more significant than unigram repetitions. However, looking at this example, it becomes clear that the repetitions found in the unigram analysis are not entirely insignificant. The phenomenon of priming explains why we repeat “the” instead of using the word “a”. It is clear that it is not by mistake that this
Repetition has a purpose in conversation, as shown by the research of Tannen, among others. Given the greater numbers of repetitions in all three original transcripts for all values of n, it would seem that repetition is not used at random by speakers. If it were, we would expect to find similar numbers of occurrences of repetition in both the original and shuffled transcripts.

N-grams had another purpose in the analysis. It was hypothesised that as n increased, the number of repetitions in the shuffled transcripts would fall significantly. The reasons for this were that it was likely that two lines in the shuffled transcript would contain words such as “the”, “a”, “and” and “to”, as these are in the top ten most common words in English. However, as n increased, there was less likelihood of finding two or more words repeated over two lines. This hypothesis proved to be true as can be seen in Section 4. In the original transcripts, the average decrease in repetitions when n was increased from 1 to 2 was 86.5%. The average decrease in repetitions in the shuffled transcripts when n was increased from 1 to 2 was 95.4%. The validity of this hypothesis was strengthened further when the value of n was 4 and 5 as, on average, there were no occurrences of repetition in the shuffled transcripts for these values. The statistical analysis of the results showed that some values of n had a significant effect on repetition, with n decreasing the mean repetition in the cases of n = 2, n = 3 and n = 4, which supports the hypothesis presented.

The independent effect of day on repetition was also found to be significant. However, the design of the program should be taken into account at this point. There were 10 shuffled transcripts created for every original transcript, meaning that there was 10 times more shuffled data. When the interactive effect of day and reality are examined, it is clear that day does not have an effect when reality is true. This means that there was no significant change in the repetition results for the original transcripts caused by the day of conversation.

These results, coupled with the theories of repetition (Tannen, Perrin et al., Rieger) and priming experiments strengthen the argument that repetition is used in a meaningful and structured way by speakers. Speakers do not repeat words in

\[\text{http://www.askoxford.com/oec/mainpage/oec02/?view=uk}\]
a random fashion, but rather they are influenced by prior utterances.

5.2.2 Overlap

Section 4 also presented two sets of results in relation to overlap. The first was overlapping speech and the second was overlapping speech with repetitions.

Two types of overlaps were examined when calculating overlap: “real” and “fake”. (see Section 3.7.3 for the reasons for differences in the results for “real” and “fake” overlap). For each day, there were more “real” and “fake” overlaps in the original transcripts than in the shuffled transcripts.

In Section 6 we saw the reasons for overlap in conversation. These were simultaneous start-ups, back-channelling and interruption. Sacks et al., Gardner and Bilmes believe turn-taking to be an organised feature of conversation and that overlaps are not produced by speakers arbitrarily. There are reasons why overlaps occur when they do.

The results of the analysis of overlap in the transcripts showed that there were more cases of overlap in the original transcript than in the shuffled transcript for each day. In fact, on day 1, there were over 10 times more occurrences of overlap in the original transcript than in the shuffled transcript. The purpose of creating a variable called “fake” overlap was to allow the shuffled transcript a chance to have overlapping utterances. Even with this chance, we can see that the number of overlaps in the original transcripts greatly outnumber those in the shuffled transcripts. The statistical analysis of the results for both “real” and “fake” overlap showed that reality had a significant effect. In fact, the p value for both conditions was identical.

The interaction of reality and day to affect overlap was interesting. One might expect that as the participants become more used to each other and to the laboratory setting, that they would tend to overlap one another more frequently. However, something different was found. The interaction of day 3 and reality = true brought down the mean overlap. This would mean that participants overlapped each other less on day 3, which is the opposite to what one might expect. However, as these were the results for “real” overlap, it was important to also look at “fake” overlap. In fact, the results for “fake” overlap indicated the same trend.
The interaction of day 2 and reality = true brought down the mean overlap as did the interaction of day 3 and reality = true. This trend indicates that speakers overlap less as they become more familiar with one another.

The overlap results found by the analysis program support the theories above which were described in detail in Section 6. The linguists mentioned above hypothesised that overlap was a structured feature of conversation and that speakers overlapped one another for non-arbitrary reasons, both positive and negative.

5.2.3 Repeated overlap

The second set of results produced by the analysis program was that of repeated overlap. Repeated overlap involves a speaker repeating the utterances of another whilst also overlapping their speech. For all values of n over all three days of conversation, there were found to be more repeated overlaps in the original transcripts than in the shuffled transcripts. On no day were repeated overlaps found in the shuffled transcripts when n was 3, 4 or 5. However, on days 1 and 2, there were 1 and 2 cases respectively of 5-gram repeated overlap. That is, speakers repeated 5 word long utterances while they overlapped the other speaker. N-gram analysis proved once again to be a valuable means of analysis. The results were then examined using R. N was found not to have a significant effect on repeated overlap. This means that speakers repeat each other independently of n, although it is clear from the results that speakers produced more unigram repeated utterances than 5gram repeated utterances.

5.3 Conclusion

All of the results which were presented in Section 4 and discussed in this section showed that reality had a significant effect on the factors of repetition, overlap and repeated overlap. These results serve as supportive evidence for the theories of overlap and repetition that were presented in Sections 2.3 and 2.4 respectively.
6 Future research

The findings of this project have many real-world applications. Three of those applications are discussed below as well as the possibility of a continuation of this project.

6.1 Continuation of this project

Due to the time constraint associated with a project of this nature, it was not possible to explore all of the possible directions that existed in relation to the topics analysed. Future extensions of this project are numerous. There is a possibility to analyse the type of words being repeated by speakers, which may give more insight into the phenomena of self-repetition and allo-repetition. There is also the possibility of exploring the area of overlap, to see if it is possible to automatically distinguish back-channelling from interruption. Another possible area of research was highlighted by the unexpected results of the significance of n on repeated overlap which was discussed in Section 5.

6.2 Artificial Intelligence

The Turing test was created by Alan Turing. The aim of the Turing test is to create a computer application which can act intelligently and during an interaction, can convince a panel of human judges that they are talking to another human rather than to a computer. During the Turing test, a human converses with a computer and is asked whether they have been talking to a human or a computer (There are also many other versions of the Turing test). Although it has its critics, the Turing test is one way in which we can look at the question of artificial intelligence and machine thinking.

The results of this project showed that priming, repetition and overlap were all features of natural conversation and that they were indications of the underlying structure of conversation. Thus, anyone designing a computer to imitate a human must take into consideration these three features of conversation. A computer which using back-channelling, interruption and similar syntactic structures may
appear more human-like than one which does not.

6.3 Film and television

As shown in Section 2, film and television program dialogue does not always conform to the structure of natural conversation. Audiences are expected to suspend belief when they watch a film or television program. Thus, they accept the dialogue despite how unnatural it might seem if taken out of the context of the film or television world. However, directors do want dialogue to seem somewhat natural, no more so than in the genre of “found footage”. “Found footage” is a genre of film-making which began in the 1980’s but was made famous by the movie “The Blair Witch Project”. “Found footage” is a type of film which is made to appear as if the footage was simply found in a video camera after the characters were killed or a disaster occurred. The events shown are advertised as being real rather than fiction.

As these types of films are intended to be seen as being real footage, directors should aim to create dialogue which imitates natural conversation as much as possible. The results of this project showed various features of conversation which occur only in the natural order of conversation. Overlapping speech and repetition would make a dialogue seem more realistic and thus convince audiences that the dialogue is real within the film.
7 Conclusion

The aim of this project was to prove that the features of natural conversation priming, repetition and overlap were features of natural conversation in its original order, thus proving that they are structured features of conversation. Having studied many linguists approaches which aimed to support that these features are structured, it was noticed that they did not provide a comparison of naturally ordered conversation and unordered conversation. They simply used naturally ordered conversation as proof of the structure of these phenomena. The experiment undertaken as part of this project analysed conversation in its natural order in comparison to the same conversations in a shuffled order.

The theories concerning the structure of conversation and the features of priming, repetition and overlap were presented in section 2. Next, the conversation analysis program created during this project was described, including the corpus used during the analysis. The results of this analysis were then presented, followed by a discussion of these results. Finally, potential uses of the results of this project were given.

The results of this project have found that repetition, overlap and priming are features of natural conversation and that they combine to show that natural conversation has an underlying structure. The use of ngrams proved to be useful as it was found that n had a significant effect on repetition. The aims of the project were completed with success and further research possibilities were found, as shown in section 6.
A Transcription conventions

The follow is an index of the meanings of all of the notation used in the examples taken from transcriptions.

. indicates falling pitch or intonation
, indicates continuing intonation
? indicates rising pitch or intonation
/?/ indicates inaudible transcription . . . indicates a pause of 1/2 second or more
(0.5) the number in parentheses indicates the length of a pause
[] indicate overlapping utterances
: indicates stretching of sound it follows
hhh indicates audible aspiration, possibly laughter
hhh indicates inbreath audible aspiration, possibly laughter
((cough)) the item within the double parentheses indicates some sound or feature of talk which is not easily transcribable
@w indicates laughter

B Analysis program code

The following is the code used to create the analysis program for this project. This is the code for bigram analysis. There are four other similar code files which contain the code for unigram, trigram, 4-gram and 5-gram analysis. Each file has slightly different code depending on the number of tokens being analysed. As all of the code is similar, only the bigram code has been included.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "readspeechdata2.h"
#define MAX 10000

/*prints an entry of type struct line*/
void printEntry(struct line* printme)
```
{
    printf("startTime:\t%s\nendTime:\t%s\nSpeaker:\t%c\nEmotion1:\t%d\nEmotion2:\t%d\n", printme->startTime, printme->endTime, printme->speaker, printme->emotion1, printme->emotion2);
    printf("Speech string: %s\n", printme->speech);
    printf("Detected tokens: %d\n", printme->num_tokens);

    int i;
    for(i = 0; i < printme->num_tokens; i++)
        printf("Token %d: %s\n", i, printme->tokenisedspeech[i]);
}

/*counts the number of lines in a file*/
int countLines(int day)
{
    int c=0;
    char ch='\0';
    FILE *fp = NULL;
    struct line* my_array;

    if(day == 1)
    {
        fp=fopen("day1.txt","r");
    }
    if(day == 2)
    {
        fp=fopen("day2.txt","r");
    }
    if(day == 3)
    {
        fp=fopen("day3.txt","r");
    }
}
while(ch!=EOF)
{
    ch=fgetc(fp);
    if(ch=='\n') c++;
}
return c+1;
}

/*generates a random number between 0 and lc*/
int randNumGenerator(int lc)
{
    srand(time(NULL));
    int randnum;

    randnum = (int)(rand()%lc);

    return randnum;
}

/*creates a new node*/
dllnode* newDllNode(int data)
{
    dllnode* newnode = NULL;
    newnode = malloc(sizeof(dllnode));
    newnode->data = data;
    newnode->prev = NULL;
    newnode->next = NULL;
    return newnode;
}

/*adds a node to a list*/
void addtolist(dllnode** headRef,int data)
{
    dllnode* node = newDllNode(data);

    //dllnode* tail = *headRef;
    node->next = *headRef;

    (*headRef)->prev = node; //here
    *headRef = node;
}
/*deletes the nth node of a list*/
void deleteNth(dllnode** headRef,int randnum)
{
    dllnode* current = *headRef;
    int i = 0;

    while(current != NULL && current->next != NULL && i != randnum)
    { /*get to the correct position in the list*/
        current = current->next;
        i++;
    }
    if (current->next != NULL && current->prev != NULL)
    {  
current->prev->next = current->next;
    current->next->prev = current->prev;
    }  
if (current->next == NULL && current->prev == NULL)
    {  
    free(current);
        return;
    }  
if (current->prev == NULL) //if the link is the head of the list

```c
{ 
    current->next->prev = NULL;
    *headRef = current->next;
}

if (current->next == NULL) //if the link is the tail of the list
{
    current->prev->next = NULL;
}

free(current);
}

/*prints a linked list*/
void printdllList(FILE* dllfile, dllnode* head)
{
    dllnode* current = head;

    if(head == NULL)
    {
        printf("Error-Empty List");
    }
    while(current!=NULL)
    {
        fprintf(dllfile,"%d ",current->data);
        current = current->next;
    }
    fprintf(dllfile,"

");
```
int getNth(int index, dllnode* head)
{
    dllnode* current = head;
    int count = 0;

    while (current != NULL)
    {
        if (count == index)
        {
            return current->data;
        }
        current = current->next;
        count++;
    }
}

int chartoint(char* time, int strlength)
{
    int output = 0;
    char newtime[9];
    int i = 0, k = 0, m = 0, min = 0, hr = 0, sec = 0, tot = 0;

    for (i = 0; i < strlength; i++)
    {
        if (time[i] != ':')
        {
            newtime[k] = time[i];
            k++;
        }
    }

newtime[k] = '\0';
for (i = 0; i < strlen(newtime); i++)
{
    output = newtime[i] - 48 + (10 * output);
}

if(output/10000 >= 1)
{
    hr = output/10000;
    tot += hr * 3600;
    output = output - (hr*10000);
}
min = output/100;
    tot += min * 60;
sec = output % 100;
tot += sec;

    return tot;  //time in seconds
} /*main function*/
int main()
{

    FILE *data = NULL;
    FILE *reps = NULL;
    FILE *shuffreps = NULL;
    FILE *day1results = NULL;
    FILE *day2results = NULL;
    FILE *day3results = NULL;
    FILE *results = NULL;
    FILE *samereps = NULL;
    FILE *samerepsrandom = NULL;
    FILE *nis2 = NULL;
FILE *overlapfile = NULL;
FILE *fakeresults = NULL;

int day = 0;
printf("Transcript day: "); /*allows user to choose day they
would like to analyse*/
scanf("%d", &day);

if (day == 1)
{
    data = fopen("day1.txt", "r");
}
if (day == 2)
{
    data = fopen("day2.txt", "r");
}
if (day == 3)
{
    data = fopen("day3.txt", "r");
}
results = fopen("linebylinerезультатs.csv", "a+"); //results file

day1results = fopen("day1results.csv", "a+");
day2results = fopen("day2results.csv", "a+");
day3results = fopen("day3results.csv", "a+");
nis2 = fopen("nis2.csv", "a+");
reps = fopen("nis2reps.txt", "a+");
shuffreps = fopen("nis2shuffreps.txt", "a+");
samereps = fopen("nis2samereps.txt", "a+");
samerepsrandom = fopen("nis2samerepsrandom.txt", "a+");
overlapfile = fopen("overlap.txt", "a+");
fakeresults = fopen("fakelinebyline.csv", "a+");
int i = 0, j = 0, z = 0, a = 0, m = 0, p = 0, sameSpeaker = 0, randnum = 0, real_random = 0;

if(data == NULL)
{
    //checking if files open correctly
    printf("Datafile could not be opened\n");
    exit(0);
}

if(day1results == NULL)
{
    printf("day1results could not be opened\n");
    exit(0);
}

if(day2results == NULL)
{
    printf("day2results could not be opened\n");
    exit(0);
}

if(day3results == NULL)
{
    printf("day3results could not be opened\n");
    exit(0);
}

int lc = countLines(day); //length of file
//Converts one line of transcript into a struct
struct line** dataArray;
if((dataArray = (struct line**)malloc(lc*sizeof(struct line*)))
  ==NULL)
    //dynamically allocate memory for dataArray
    {
      printf("Not enough memory!");
      //make sure it allocated correctly
      exit(0);
    }

int lineCount = 0;

for(i = 0; i < lc; i++)
    dataArray[i] = NULL;
  //fill dataArray with null characters

char temp[1000];

rewind(data);
  //make sure cursor is at the beginning of the file

while(!feof(data))
  {
    // allocate new item
    dataArray[lineCount] = (struct line*)malloc(sizeof(struct line));
    // scan current line
    fgets(temp, 1000, data);
    i = 0;
    j = 0;
// copy start time into struct
while(temp[i] != ' ' && temp[i] != '	'){
    dataArray[lineCount]->startTime[j++] = temp[i++];
}
dataArray[lineCount]->startTime[j] = '\0';

while(temp[i] == ' ' || temp[i] == '	')
i++;
j = 0;

// copy end time into struct
while(temp[i] != ' ' && temp[i] != '	'){
    dataArray[lineCount]->endTime[j++] = temp[i++];
}
dataArray[lineCount]->endTime[j] = '\0';

while(temp[i] == ' ' || temp[i] == '	')
i++;
j = 0;

// fill in speaker char
dataArray[lineCount]->speaker = temp[i++];

// skip over PLAY
while(!(temp[i] >= '0' && temp[i] <= '9'))
i++;

// copy emotion1 into struct
dataArray[lineCount]->emotion1 = (int)temp[i++] - 0x30;
while(temp[i] == ' ' || temp[i] == '\t')
    i++;
//copy emotion2 into struct
dataArray[lineCount]->emotion2 = (int)temp[i++] - 0x30;
while(temp[i] == ' ' || temp[i] == '\t')
    i++;

strcpy(dataArray[lineCount]->speech, &temp[i]);

dataArray[lineCount]->num_tokens = 0;
//tokenise speech
char* pch;
pch = strtok(dataArray[lineCount]->speech,
" \t\n,?!"(){});
while(pch != NULL)
{
    strcpy(&dataArray[lineCount]->tokenisedspeech[
dataArray[lineCount]
    ->num_tokens++], pch);  
pch = strtok(NULL, " \t\n,?!"(){}");
}

lineCount++;

//variables used during analysis

83
int tokens_total = 0, tokens_shared = 0, bin_same_speaker = 0, tokensintext = 0;
int repoverlap = 0, yes = 0, overlapcheck = 0, totaloverlap = 0, time = 0,
ntotal = 0, ytotal = 0, ktotal = 0, gtotal = 0, dtotal = 0;
float ntime = 0, ktime = 0, gtime = 0, ytime = 0, dtime = 0, floattime = 0;
int overlaptime = 0, ot = 0, repoverlapcounter = 0, repoverlaptime = 0,
orderedSentences = 0;
int randomsti = 0, randomsti1 = 0, randometi = 0, randometi1 = 0;
int fakeoverlaptime = 0, fakerepoverlap = 0, totalfakeoverlaptime = 0,
fakeoverlapcheck = 0, totalfakeoverlap = 0;
int f = 0, fakerepcounter = 0, fakerepoverlaptime = 0, bin_allo = 0, bin_rep = 0;
int newSTi = 0, newETi = 0; // start time and end time of line i as integers
int newSTi1 = 0, newETi1 = 0; // start and end time of line i+1 as integers

/*/analysing dataArray line by line for matching words and matching speakers*/
char* string1 = NULL;
char* string2 = NULL;
i = 0;
j = 0;
k = 0;
int count = 0;
while(i < lineCount-1)
{
    if(dataArray[i+1]->num_tokens > 1 && dataArray[i]->num_tokens > 1)
    {
        orderedSentences += dataArray[i]->num_tokens - 1;
        while(k < dataArray[i+1]->num_tokens - 1)
        {
            // as this is n = 2, the program combines 2 words to make a token
            string2 = (char*)malloc((strlen(dataArray[i+1]
                ->tokenisedspeech[k]) + strlen(dataArray[i+1]
                ->tokenisedspeech[k+1]) + 2) * sizeof(char));
        }
    }
strcpy(string2, dataArray[i+1]->tokenisedspeech[k]);
strcat(string2, " ");
strcat(string2, dataArray[i+1]->tokenisedspeech[k+1]);

while (j < dataArray[i]->num_tokens - 1)
{
    if (string1 != NULL)
    {
        free(string1);
    }
    string1 = (char*)malloc((strlen(dataArray[i]->tokenisedspeech[j]) +
        strlen(dataArray[i]->tokenisedspeech[j+1]) + 2)*sizeof(char));
    strcpy(string1, dataArray[i]->tokenisedspeech[j]);
    strcat(string1, " ");
    strcat(string1, dataArray[i]->tokenisedspeech[j+1]);

    // checks if the tokens from line i are the same as line i+1
    if (strcmp(string1, string2) == 0)
    {
        tokens_shared++;
        count++;
        yes = 1; // a value of 1 indicates repetition
    }

    // checks if there is self-repetition
    if (dataArray[i]->speaker == dataArray[i+1]->speaker)
    {
        bin_same_speaker = 1;
        samespeaker++;
        fprintf(samereps, "%s\t\n", dataArray[i]->tokenisedspeech[j]);
    }
}

85
j++;
}
k++;
j=0;
free(string2);
}
real_random = 0; /*reality value is 0 indicating that this is the
real order*/
//calculates the total number of tokens between the two lines
if((dataArray[i]->num_tokens > 1) && (dataArray[i+1]->num_tokens > 1))
{
tokens_total = dataArray[i]->num_tokens + dataArray[i+1]->num_tokens - 2;
//n = 2,1 less token per line
}
else
{
tokens_total = 0;
}
char *et,*st;
et = dataArray[i]->endTime;
st = dataArray[i]->startTime;
int etlen = strlen(dataArray[i]->endTime);
int stlen = strlen(dataArray[i]->startTime);
int newet = chartoint(et,etlen);
int newst = chartoint(st,stlen);
time = newet - newst;
if (time == 0)
{
time = 1; /*if an utterance has the same start and end time, they lasted
for a second, not 0 seconds*/
}
// calculates total time of speech by speaker
float time = (float)time;
if (dataArray[i]->speaker == 'd')
{
    dtotal += dataArray[i]->num_tokens;
dtime += floattime;
}
if (dataArray[i]->speaker == 'k')
{
    ktotal += dataArray[i]->num_tokens;
ktime += floattime;
}
if (dataArray[i]->speaker == 'g')
{
    gtotal += dataArray[i]->num_tokens;
gtime += floattime;
}
if (dataArray[i]->speaker == 'n')
{
    ntotal += dataArray[i]->num_tokens;
ntime += floattime;
}
if (dataArray[i]->speaker == 'y')
{
    ytotal += dataArray[i]->num_tokens;
ytime += floattime;
}

newSTi = chartoint(dataArray[i]->startTime,
strlen(dataArray[i]->startTime));
newETi = chartoint(dataArray[i]->endTime, strlen(dataArray[i]->endTime));
newSTi1 = chartoint(dataArray[i+1]->startTime, strlen(dataArray[i+1]->startTime));
newETi1 = chartoint(dataArray[i+1]->endTime, strlen(dataArray[i+1]->endTime));

char fakeeti[3];
char fakeeti1[3];
char fakesti[3];
char fakesti1[3];
a = 0;
f = 0;
//creates fake times for fakeoverlap
for(a = 5; a < strlen(dataArray[i]->startTime); a++)
{
    fakesti[f] = dataArray[i]->startTime[a];
f++;
}
fakeeti[f] = '\0';
f = 0;
for(a = 5; a < strlen(dataArray[i]->endTime); a++)
{
    fakeeti[f] = dataArray[i]->endTime[a];
f++;
}
fakeeti[f] = '\0';
f = 0;
for(a = 5; a < strlen(dataArray[i+1]->startTime); a++)
{
    fakesti1[f] = dataArray[i+1]->startTime[a];
f++;
}
fakestil[f] = '\0';
f = 0;
for(a = 5; a < strlen(dataArray[i+1]->endTime); a++)
{
    fakeeti1[f] = dataArray[i+1]->endTime[a];
f++;
}
fakeeti1[f] = '\0';

randomsti = chartoint(fakesti,strlen(fakesti));
randomsti1 = chartoint(fakesti1,strlen(fakesti1));
randometi = chartoint(fakeeti,strlen(fakeeti));
randometi1 = chartoint(fakeeti1,strlen(fakeeti1));
// checks for fake overlap
if (((randomsti == randomsti1) || (randometi == randometi1))
{
    if(yes == 1) // there is repetition
    {
        fakerepoverlap = 1;
fakerepcounter++;
    }
totfaakeoverlap++;
    fakeoverlapcheck = 1;
    if(randomsti == randomsti1 && randometi == randometi1)
    {
        ot = randometi - randomsti; // time spent overlapping
    }
    if(randomsti == randomsti1 && randomsti != randometi1)
    {
        if(randometi > randometi1)
        {

89
ot = randometi1 - randomstil;
}
else
{
  ot = randometi - randomstil;
}
}

if (randometi == randometi1 && randomstil != randomstil1)
  {
    if (randomstil > randomstil1)
      {
        ot = randometi - randomstil;
      }
    else
      {
        ot = randometi1 - randomstil1;
      }
  }

if ((fakeoverlapcheck == 1) && (ot == 0))
  {
    totalfakeoverlap--;
    fakeoverlapcheck = 0;
  }

if ((fakerepoverlap == 1) && (ot == 0))
  {
    fakerepcounter--;
    fakerepoverlap = 0;
  }

if ((fakeoverlapcheck == 1) && ((randometi - randomstil == 0) || (randometi1 - randomstil1 == 0)))
  {
    totalfakeoverlap--;
  }
fakeoverlaptime += ot;
if(fakerepoverlap == 1)
{
  fakerepoverlaptime += ot;
}

if ((randomsti1 > randomsti && randomsti1 < randometi
  && randometi1 != randometi)|| (randometi1 > randomsti
  && randometi1 < randometi && randomsti1 != randomsti))
{
  if(yes == 1)
  {
    fakerepoverlap = 1;
  }
  fakeoverlapcheck = 1;
totalfakeoverlap++;

if(randometi1 < randometi && randometi1>randomsti)
{
  ot = randometi1 - randomsti1;
}
if(randometi1 > randometi)
{
  ot= randometi - randomsti1;
}
  if(randomsti1 < randomsti)
  {
    ot= randometi1 - randomsti;
  }
  if((fakeoverlapcheck == 1) && (ot == 0))
totalfakeoverlap--; 
fakeoverlapcheck = 0;
}
if((fakerepoverlap == 1) && (ot == 0))
{
fakerepcounter--; 
fakerepoverlap = 0;
}
if((fakeoverlapcheck == 1) && (((randometi - randomsti == 0) 
|| (randometi1 - randomsti1 == 0))))
{
totalfakeoverlap--; 
}
fakeoverlaptime += ot;
if(fakerepoverlap == 1)
{
fakerepoverlaptime += ot; 
}
//calculates real overlap
if (((newSTi == newSTi1) || (newETi == newETi1))
{
if(yes == 1)
{
repoverlap = 1;
repoverlapcounter++; 
}
totaloverlap++; 
overlapcheck = 1;
if(newSTi == newSTi1 && newETi == newETi1)
{
ot = newETi - newSTi;
}
if(newSTi == newSTi1 && newETi != newETi1)
{
 if(newETi > newETi1)
{
    ot = newETi1 - newSTi1;
}
else
{
    ot = newETi - newSTi;
}
}
if (newETi == newETi1 && newSTi != newSTi1)
{
 if(newSTi > newSTi1)
{
    ot = newETi - newSTi;
}
else
{
    ot = newETi1 - newSTi1;
}
}
if((overlapcheck == 1) && (ot == 0))
{
    totaloverlap--;
    overlapcheck = 0;
}
    if((reoverlap == 1) && (ot == 0))
    {
        reoverlapcounter--;
repoverlap = 0;
}
overlaptime += ot;
if(repoverlap == 1)
{
overlaptime += ot;
}
}
if ((newSTi1 > newSTi && newSTi1 < newETi && newETi1 != newETi) ||
(newETi1 > newSTi && newETi1 < newETi && newSTi1 != newSTi))
{
   if(yes == 1)
   {
      repoverlap = 1;
      repoverlapcounter++;
   }
overlapcheck = 1;
totaloverlap++;

   if(newETi1 < newETi && newSTi1>newSTi)
   {
      ot = newETi1 - newSTi1;
   }
if(newETi1 > newETi)
{
   ot= newETi - newSTi1;
}
if(newSTi1 < newSTi)
{
   ot= newETi1 - newSTi;
}
if((overlapcheck == 1) && (ot == 0))
{ totaloverlap--; }

if((repoverlap == 1) && (ot == 0))
{
    repoverlapcounter--; }

if(repoverlap == 1)
{
    repoverlapt ime += ot;
}
overlaptime += ot;

if (overlapcheck == 1)
{
    fprintf(overlapfile,"%d,%d\n",i,ot);
}

if((tokens_shared > 0) && (bin_same_speaker == 0))
{
    bin_allo = 1;
}

if((bin_same_speaker == 1) || (bin_allo == 1))
{
    bin_rep = 1;
}

//prints the results for the pair of lines analysed
fprintf(results,"%d,2,%d,%c,%c,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d
",
    day,i, dataArray[i]->speaker, dataArray[i+1]->speaker, bin_same_speaker ,
    bin_allo, bin_rep, tokens_shared, tokens_total, real_random,
    overlapcheck, repoverlap, fakeoverlapcheck, fakerepoverlap);
//resets all variables
bin_same_speaker = 0;
tokens_shared = 0;
tokens_total = 0;
repoverlap = 0;
bin_allo = 0;
bin_rep = 0;
overlapcheck = 0;
fakerepoverlap = 0;
fakeoverlapcheck = 0;

i++; //analyse next line
k=0;
yes = 0;
}

//terminal print-outs for user
printf("amount of repetition in real data: %d",count);
printf("\nsself repetition in real data: %d\n",samespeaker);
printf("\ngtotal: %d : ktotal: %d : ntotal: %d : ytotal: %d :
dtotal: %d",gtotal,ktotal
,ntotal,ytotal,dtotal);
float tot = gtotal + ktotal + ntotal +ytotal +dtotal;
printf("\nwords: g: %f : k: %f : n: %f : y: %f : d: %f",
gtotal/tot,ktotal/tot,ntotal/tot,ytotal/tot,
dtotal/tot,tot,tot,tot);
printf("\ngtime: %f : ktime: %f :ntime: %f :ytime: %f :
dtime: %f",gtime,ktime,ntime,ytime,dtime);
floattime = gtime + ktime + ntime + ytime + dtime;
printf("\nntime: g: %f : k: %f : n: %f : y: %f : d: %f",
gtime/floattime,ktime/floattime,ntime/floattime,ytime/floattime,dtime/floattime);
printf("\n\ntotal overlap: %d",totaloverlap);
printf("\noverlaptime: %d",overlaptime);
printf("\n\nrepeated overlap: %d", repoverlapcounter);
printf("\n\nrepeated overlap time: %d", repoverlaptime);
printf("\n\nfake overlap: %d", totalfakeoverlap);
printf("\n\nfake overlap time: %d", fakeoverlaptime);

//creates new variables for shuffled file analysis
int totalCount = 0;
int totalSpeaker = 0;
int average = 0;
int speakeravg = 0;
int times = 0;
int randomSentences = 0;
int randomoverlap = 0;

//resets all variables
tokens_shared = 0;
tokens_total = 0;
real_random = 0;
bin_same_speaker = 0;
reoverlap = 0;
overlapcheck = 0;
totaloverlap = 0;
yes = 0;
bin_allo = 0;
fakeoverlaptime = 0;
fakereoverlap = 0;
totalfakeoverlaptime = 0;
fakeoverlapcheck = 0;
totalfakeoverlap = 0;
fakereoverlaptime = 0;
fakerepcounter = 0;
reoverlapcounter = 0;

//creation of shuffled transcripts
//an array of shuffledLine struct pointers
struct shuffledLine** shuffledData;
if((shuffledData = (struct shuffledLine**)malloc(lc*sizeof(struct shuffledLine*)))==NULL)
{
    printf("Not enough memory!");
    exit(0);
}

real_random = 1;
while(times < 10) //number of shuffled transcripts to create
{
    //bags approach begins
    /*creates a linked list of length lc which is filled
    with the indices 0->lc*/
    int index = 1,n = 0,random = 0,s = 0,counter = 0,y=0;
    dllnode* node = NULL;
    node = (dllnode*)malloc(sizeof(dllnode));
    node->prev = NULL;
    node->next = NULL;
    node->data = 0;

    while(index<lc)
    {
        addtolist(&node,index);
        index++;
    }
    index = 0;

    for(i = 0; i < lc; i++)
    {
        shuffledData[i] = NULL;
        shuffledData[i] = (struct shuffledLine*)malloc(sizeof

(struct shuffledLine));
}
y = lc;

while(index<lc)
{
    shuffledData[index]->newIndex = index;
    shuffledData[index]->oldIndex = index;
    index++;
}

/*generates a random number, gets the index n at the node at position of the random number generated. gives the struct at index of shuffledData array a new index of index and an old index of random. Then deletes the node at position random (so that it is not used again.decrements y because linked list is one element smaller now.*/

index = 0;
while(index<lc)
{
    if (y > 1)
    {
        random = randNumGenerator(y);
    }
    else
    {
        random = 0;
    }
    n = getNth(random,node);
    shuffledData[index]->newIndex = index;
    shuffledData[index]->oldIndex = n;
index++;  
y--;  
deleteNth(&node,random);
}

i = 0;  
j = 0;  
int random_samespeaker = 0;  
k = 0;  
int random_count = 0;  

/*as earlier analysis but for shuffled order this time*/
while(i < lc-1)  
{
    if((dataArray[shuffledData[i+1]->oldIndex]->num_tokens > 1) &&
    (dataArray[shuffledData[i]->oldIndex]->num_tokens >1))
    {
        randomSentences+= dataArray[shuffledData[i]->oldIndex]->num_tokens - 1;
        while(k < dataArray[shuffledData[i+1]->oldIndex]->num_tokens-1)
        {
            string2 = (char*)malloc((strlen(dataArray[shuffledData[i+1]
->oldIndex]->tokenisedspeech[k])+ strlen(dataArray[
shuffledData[i+1]->oldIndex]->tokenisedspeech[k+1])+2)
*sizeof(char));
            strcpy(string2,dataArray[shuffledData[i+1]->oldIndex]
->tokenisedspeech[k]);
            strcat(string2," ");
            strcat(string2,dataArray[shuffledData[i+1]->oldIndex]
->tokenisedspeech[k+1]);
        }
    }
}
while (j < dataArray[shuffledData[i]->oldIndex]->num_tokens-1) {
    if (string1 != NULL ) {
        free(string1);
    }
    string1 = (char*)malloc((strlen(dataArray[shuffledData[i]->oldIndex]->tokenisedspeech[j])+strlen(dataArray[shuffledData[i]->oldIndex]->tokenisedspeech[j+1])+2)*sizeof(char));
    strcpy(string1, dataArray[shuffledData[i]->oldIndex]->tokenisedspeech[j]);
    strcat(string1, " ");
    strcat(string1, dataArray[shuffledData[i]->oldIndex]->tokenisedspeech[j+1]);

    if (strcmp(string1, string2) == 0) {
        tokens_shared++;
        yes = 1;
        fprintf(shuffreps,"%s\n", dataArray[shuffledData[i]->oldIndex]->tokenisedspeech[j]);
        random_count++;

        if (dataArray[shuffledData[i]->oldIndex]->speaker == dataArray[shuffledData[i+1]->oldIndex]->speaker) {
            bin_same_speaker = 1;
            random_samespeaker++;
        }
    }
    j++;
}
k++; 
j=0; 
free(string2); 
}
}
if((dataArray[shuffledData[i]->oldIndex]->num_tokens) > 1 &&
(dataArray[shuffledData[i+1]->oldIndex]->num_tokens) > 1)
{
tokens_total = dataArray[shuffledData[i]->oldIndex]->num_tokens +
dataArray[shuffledData[i+1]->oldIndex]->num_tokens - 2;
}
else
{
tokens_total = 0;
}
newSTi = chartoint(dataArray[shuffledData[i]->oldIndex]->startTime,
strlen(dataArray[shuffledData[i]->oldIndex]->startTime));
newETi = chartoint(dataArray[shuffledData[i]->oldIndex]->endTime,
strlen(dataArray[shuffledData[i]->oldIndex]->endTime));
newSTi1 = chartoint(dataArray[shuffledData[i+1]->oldIndex]->startTime,
strlen(dataArray[shuffledData[i+1]->oldIndex]->startTime));
newETi1 = chartoint(dataArray[shuffledData[i+1]->oldIndex]->endTime,
strlen(dataArray[shuffledData[i+1]->oldIndex]->endTime));

char fakeeti[3];
char fakeeti1[3];
char fakesti[3];
char fakesti1[3];
a = 0;
f = 0;

for(a = 5;a < strlen(dataArray[shuffledData[i]->oldIndex]->startTime);a++)
fakesti[f] = dataArray[shuffledData[i]->oldIndex]->startTime[a];
    f++;
}
fakesti[f] = '\0';
f = 0;
for(a = 5;a < strlen(dataArray[shuffledData[i]->oldIndex]->endTime);a++)
{
    fakeeti[f] = dataArray[shuffledData[i]->oldIndex]->endTime[a];
    f++;
}
fakeeti[f] = '\0';
f = 0;
for(a = 5;a < strlen(dataArray[shuffledData[i+1]->oldIndex]->startTime);a++)
{
    fakesti1[f] = dataArray[shuffledData[i+1]->oldIndex]->startTime[a];
    f++;
}
fakesti1[f] = '\0';
f = 0;
for(a = 5;a < strlen(dataArray[shuffledData[i+1]->oldIndex]->endTime);a++)
{
    fakeeti1[f] = dataArray[shuffledData[i+1]->oldIndex]->endTime[a];
    f++;
}
fakeeti1[f] = '\0';

randomsti = chartoint(fakesti,strlen(fakesti));
randomstil = chartoint(fakesti1,strlen(fakesti1));
randometi = chartoint(fakeeti,strlen(fakeeti));
randometil = chartoint(fakeeti1,strlen(fakeeti1));
if ((randomsti == randomstil) || (randometi == randometil))
{ if(yes == 1) {
  fakerepoverlap = 1;
  fakerepcounter++;
}
totalfakeoverlap++;
fakeoverlapcheck = 1;
if(randomsti == randomsti1 && randometi == randometi1) {
  ot = randometi - randomsti;
}
if(randomsti == randomsti1 && randomsti != randometi1) {
  if(randometi > randometi1) {
    ot= randometi1 - randomsti1;
  } else {
    ot= randometi - randomsti;
  }
} else {
  ot= randometi - randomsti;
}
}
if (randometi == randometi1 && randomsti != randomsti1) {
  if(randomsti > randomsti1) {
    ot= randometi - randomsti;
  } else {
    ot= randometi1 - randomsti1;
  }
}
fakeoverlap += ot;
if((fakeoverlapcheck == 1) && (ot == 0))
{
    totalfakeoverlap--;
    fakeoverlapcheck = 0;
}
if((fakerepoverlap == 1) && (ot == 0))
{
    fakerepcounter--;
    fakerepoverlap = 0;
}
if((fakeoverlapcheck == 1) && ((randometi - randomsti == 0) ||
(randometi1 - randomsti1 == 0)))
{
    totalfakeoverlap--;
}

if ((randomsti1 > randomsti && randomsti1 < randometi &&
randometi1 != randometi) || (randometi1 > randomsti &&
randometi1 < randometi && randomsti1 != randomsti))
{
    if(yes == 1)
    {
        fakerepoverlap = 1;
        fakerepcounter++;
    }
    fakeoverlapcheck = 1;
    totalfakeoverlap++;
if(randometi1 < randometi1 && randomsti1 > randomsti1)
{
    ot = randometi1 - randomsti1;
}
if(randometi1 > randometi)
{
    ot = randometi1 - randomsti1;
}
if(randomsti1 < randomsti)
{
    ot = randometi1 - randomsti1;
}

fakeoverlaptime += ot;
if(((fakeoverlapcheck == 1) && (ot == 0))
{
    totalfakeoverlap--;
    fakeoverlapcheck = 0;
}
if((fakerepoverlap == 1) && (ot == 0))
{
    fakerepcounter--;
    fakerepoverlap = 0;
}
if(((fakeoverlapcheck == 1) && ((randometi - randomsti == 0) || (randometi1 - randomsti1 == 0))))
{
    totalfakeoverlap--;
}
if ((newSTi == newSTi1) || (newETi == newETi1))


if(yes == 1)
{
    repoverlap = 1;
    repoverlapcounter++;
}
totaloverlap++;
overlapcheck = 1;
    if(newSTi == newSTi1 && newETi == newETi1)
        {
            ot = newETi - newSTi;
        }
    if(newSTi == newSTi1 && newETi != newETi1)
    {
        if(newETi > newETi1)
        {
            ot = newETi1 - newSTi1;
        }
        else
        {
            ot = newETi - newSTi;
        }
    }
    if (newETi == newETi1 && newSTi != newSTi1)
    {
        if(newSTi > newSTi1)
        {
            ot = newETi - newSTi;
        }
        else
        {
            ot = newETi1 - newSTi1;
        }
    }
overlaptime += ot;
if((ot == 0) && (overlapcheck == 1))
{
    totaloverlap--;
    overlapcheck = 0;
}
if((repoverlap == 1) && (ot == 0))
{
    repoverlapcounter--;
    repoverlap = 0;
}
if(repoverlap == 1)
{
    repoverlaptime += ot;
}

if ((newSTi1 > newSTi && newSTi1 < newETi && newETi1 != newETi) ||
    (newETi1 > newSTi && newETi1 < newETi && newSTi1 != newSTi))
{
    if(yes == 1)
    {
        repoverlap = 1;
        repoverlapcounter++;
    }
    overlapcheck = 1;
    totaloverlap++;

    if(newETi1 < newETi && newSTi1>newSTi)
    {

ot = newETi1 - newSTi1;
}
if(newETi1 > newETi) {
    ot = newETi - newSTi1;
}
if(newSTi1 < newSTi) {
    ot = newETi1 - newSTi;
}

overlaptime += ot;
if((ot == 0) && (overlapcheck == 1)) {
    totaloverlap--; overlapcheck = 0;
}
if((repoverlap == 1) && (ot == 0)) {
    repoverlapcounter--; repoverlap = 0;
}
if(repoverlap == 1) {
    repoverlaptime += ot;
}

if((tokens_shared > 0) && (bin_same_speaker == 0)) {
    bin_allo = 1;
}
if((bin_same_speaker == 1) || (bin_allo == 1))
{ 
    bin_rep = 1;
}

fprintf(results, "%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d,%d\n",
    day,i,dataArray[shuffledData[i]->oldIndex]->speaker,dataArray
    [shuffledData[i+1]->oldIndex]->speaker,bin_same_speaker,bin_allo,
    bin_rep,tokens_shared,tokens_total,real_random,
    overlapcheck,repoverlap,fakeoverlapcheck,fakerepoverlap);
    bin_same_speaker = 0;
    tokens_shared = 0;
    bin_allo = 0;
    tokens_total = 0;
    bin_rep = 0;
    overlapcheck = 0;
    repoverlap = 0;
    fakerepoverlap = 0;
    yes = 0;
    i++;
    k=0;
}

totalCount += random_count; /*add values from this shuffled transcript
to total*/
totalSpeaker += random_samespeaker;
times++; //next shuffled file
real_random++; //number of shuffled file being analysed
randomoverlap += overlaptime;
totalfakeoverlaptime += fakeoverlaptime;
fakeoverlaptime = 0;
}
//calculate averages
randomoverlap = randomoverlap/times;
totalfakeoverlap = totalfakeoverlap/times;
printf("\ntotalCount: %d",totalCount);
printf("\ntotalSpeaker: %d",totalSpeaker);
average = totalCount/times;
speakeravg = totalSpeaker/times;
printf("amount of repetition in shuffled file: %d",average);
printf("\nself repetition in shuffled file: %d",speakeravg);
printf("\namount of randomised sentences with n > 2: %d",
randomSentences/times);
printf("\namount of ordered sentences with n > 2: %d",
orderedSentences);
printf("\nfakerepoverlap: %d",fakerepcounter/10);
if(day == 1)
{
    fprintf(day1results,"\n2,%d,%d,%d,%d",count,samespeaker,
average,speakeravg);
}
if(day == 2)
{
    fprintf(day2results,"\n2,%d,%d,%d,%d",count,samespeaker,
average,speakeravg);
}
if(day == 3)
{
    fprintf(day3results,"\n2,%d,%d,%d,%d",count,samespeaker,
average,speakeravg);
}
fprintf(nis2,"\n%d,%d,%d,%d",day,count,samespeaker,
average,speakeravg);
fclose(data);       // close data files
fclose(reps);
    fclose(shuffreps);
    fclose(results);
    fclose(day1results);
    fclose(day2results);
    fclose(day3results);
    fclose(nis2);
    fclose(samereps);
    fclose(samerepsrandom);

    return 0;
}
References


113


