Short Papers

Short papers presented as posters at the
2017 Modern AI and Cognitive Science Conference.

A study of how people infer social relationships from people's behavior in simple economic games

Nathan Blank
Rose-Hulman Institute of Technology
Dept. of Comp. Sci. and Software Eng.
blanknc@rose-hulman.edu

Charles Kemp
Carnegie Mellon University
Dept. of Psychology
ckemp@cmu.edu

Alan Jern
Rose-Hulman Institute of Technology
Dept. of Humanities and Social Sciences
jern@rose-hulman.edu

ABSTRACT

We explore several models of social reasoning to better understand how people make inferences about people's social relationships after observing their behavior. In an experiment, we find that there are individual differences in subjects' social inferences and no single model accounts well for most subjects' inferences.

1 INTRODUCTION

After observing two people interact, it is often possible for an observer to infer what kind of relationship the two people have. In this paper, we explore the mental representations that support these inferences. Previous studies have suggested that people represent other people as rational agents in order to reason about their mental states, choices, and relationships with other people [1, 3]. In this paper, we address whether this holds when people are making inferences about social interactions. Specifically, we focus on situations in which a person observes two people simultaneously making choices that affect themselves and the other person; the observer then infers whether the pair are friends, strangers, or enemies.

We studied this question using simple economic games, like the prisoner's dilemma. Each game has two players, each with two options. We chose these games because they represent stereotypical social interactions [4]. The two players' choices in a game can be used to infer their relationship. For example, in the prisoner's dilemma, if both players cooperate, one might infer that the players trust each other, and are friends. If the players both act greedily and defect, one might infer that the players do not trust each other and are enemies or strangers. We ran an experiment in which subjects made inferences like these. We compared subjects' judgments to the predictions of three different computational models to better understand how they made these inferences [1].

2 MODELS

The three models varied in complexity and number of assumptions. We will describe the models in order of decreasing complexity.

2.1 Recursive model

The recursive model first assumes that observers expect players to make choices based on the payouts they and the other player get in the game. We capture this by assigning weights to how much a player cares about her own payout and how much she cares about the other player's payout (cf. [2]). Specifically, we represent how much Player A cares about her own payout as w_{AA} and how much Player A cares about Player B's payout as w_{AB}, and similarly for Player B. These weights are constrained so that $w_{AA} + w_{AB} = 1$. We represent whether Player A wants Player B to receive positive or negative payout as $\gamma_{AB} \in \{-1, +1\}$.

These parameters can capture different relationships. For friends, we assume that $\gamma_{AB} = +1$ and $w_{AB} > w_{AA}$. For enemies, $\gamma_{AB} = -1$, and $w_{AB} > w_{AA}$. For strangers, we assume that the players care more about themselves than the other player but that they want to help the other player: $\gamma_{AB} = +1$ and $w_{AA} > w_{AB}$. These assumptions have previous empirical support [3, 5].

To perform inferences about two players' relationship, we compute the probability of each relationship (setting of parameters) given an observed outcome in a game. For example, to find the probability that two players are friends, we iterate over settings of our parameters (i.e., all weights subject to the $w_{AB} > w_{AA}$ constraint). For each setting of parameters, we compute the probability that the players would make the observed choices, assuming a softmax utility function. We compute the mean probability of making the observed choice averaged over all possible parameter settings. We repeat this procedure for strangers and enemies. We then normalize the resulting mean probabilities to produce a probability distribution over the three relationships.

We call this model the recursive model because it assumes that observers of a game assume that each player takes the other player's choice into account before making a choice. For example, if two players are friends, the model assumes that each player expects the other player to choose using the parameter settings for friends. For simplicity, we stopped this recursive reasoning at a depth of one. That is, each player assumes that the other player is modeling them in return as a random agent.

2.2 Independent agents model

The independent agents model is similar to the recursive model, but does not assume that the players take the other players' choices into account. Instead, this model assumes that each player assigns a utility to each of the four possible game outcomes, based on the payouts and the player's relationship with the other player. We then assign probabilities to each outcome in proportion to the player's utility. We repeat this procedure for the other player. We then combine the probabilities for both players by multiplying them for each outcome, and then normalizing so that the total probability for all outcomes in a game sums to 1.

2.3 Heuristic model

It is possible that people make social inferences by relying on simple cues. To test this possibility, we also created a heuristic model. This model predicts that the players are friends if they end up in an outcome with the total maximum payout for both players, enemies

[1] All modeling code, materials, and data are available at https://github.com/jernlab/social-reasoning

Figure 1: *All subjects' ratings. Conditions on the x-axis are ordered by mean rating assigned to friends, from lowest (left) to highest (right).*

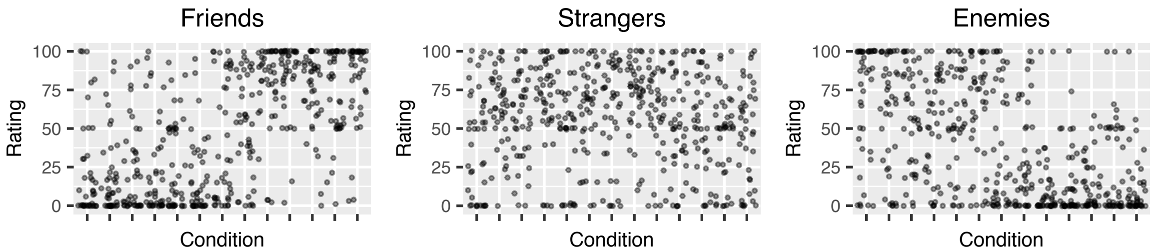
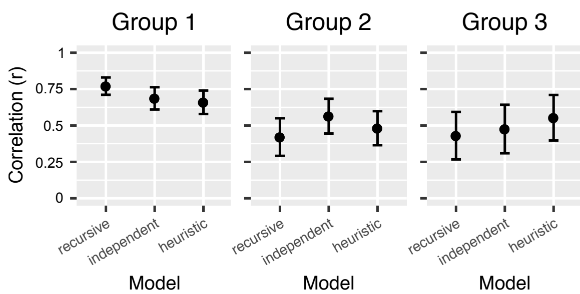


Figure 2: *Mean correlations between subjects' ratings and model predictions, for three different groups of subjects. Error bars are 95% confidence intervals.*



if they end up in an outcome with the total minimum payout, and strangers otherwise.

3 EXPERIMENT

We conducted an experiment to test what inferences people actually make. Subjects were 60 users on Amazon Mechanical Turk, 20 of which were excluded for failing an attention check (instructions to only enter a '0' in a text box). There were 13 within-subjects conditions. In each condition, subjects saw a table of the game's payouts with the players' outcome in the game identified. Subjects were told that the players were not able to communicate in the game. In total, we used five different games and between two and four different outcomes for each game. The conditions were shown in random order, but conditions involving the same game (with different outcomes) were shown sequentially before moving onto a new game. Subjects rated how likely it was that the players were friends, strangers, and enemies using three sliders that ranged from 0 ("very unlikely") to 100 ("very likely"). Finally, subjects were asked to explain their judgments in a text box.

4 RESULTS AND DISCUSSION

Figure 1 shows all subjects' ratings for all conditions, ordered by mean rating for friends from lowest to highest. Subjects' ratings were highly inconsistent for strangers, compared to friends and enemies. Examining individual subject's ratings and written explanations revealed that some subjects expected strangers to act selfishly, some expected strangers to act ambivalently, and many subjects assigned high ratings to strangers when the outcome of the game was difficult to make sense of otherwise. One example: "In this case it is almost impossible to determine, since all the choices are the same except for [outcome 4], but even enemies wouldn't pick that one because they would be sabotaging themselves as well". Our data therefore suggest that people's intuitions about how friends and enemies behave toward one another are more consistent than their intuitions about strangers.

Due to the inconsistency in responses about strangers, we further analyzed data and predictions only for friends and enemies. We identified the best-fitting model for each subject by computing the Pearson's correlation coefficient between each model and each subject's ratings. Figure 2 shows the mean correlation coefficients for subjects in three groups that resulted from this analysis. Group 1 ($N = 9$) includes subjects that were best fit by the recursive model ($r = 0.77$) compared to the independent model ($r = 0.69$) and the heuristic model ($r = 0.66$). Group 2 ($N = 19$) includes subjects that were best fit by the independent agent model ($r = 0.56$) compared to the recursive model ($r = 0.42$) and the heuristic model ($r = 0.48$). Group 3 ($N = 12$) includes subjects that were best fit by the heuristic model ($r = 0.55$) compared to the recursive model ($r = 0.43$) and the independent agent model ($r = 0.47$).

None of our models provided a strong account of all subjects' judgments, suggesting that people may approach our task in qualitatively different ways. However, about three quarters of our subjects were best-fit by either the recursive or independent agents models—both of which include models of the players' decision-making behavior—rather than the heuristic model, which does not. This result suggests that most people reason about social relationships by modeling the behavior of the people involved, rather than by relying on superficial heuristic cues.

REFERENCES

[1] C. L. Baker and J. B. Tenenbaum. 2014. Modeling human plan recognition using Bayesian theory of mind. In *Plan, activity and intent recognition: Theory and practice*, G. Sukthankar, R. P. Goldman, C. Geib, D. Pynadath, and H. Bui (Eds.). Morgan Kaufmann.

[2] D. W. Griesinger and J. W. Livingston, Jr. 1973. Toward a model of interpersonal motivation in experimental games. *Behavioral Science* 18, 3 (1973), 173–188.

[3] Alan Jern and Charles Kemp. 2014. Reasoning about social choices and social relationships. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*.

[4] Harold H. Kelley, John G. Holmes, Norbert L. Kerr, Harry T. Reis, Caryl E. Rusbult, and Paul A. M. Van Lange. 2002. *An atlas of interpersonal situations*. Cambridge University Press.

[5] G. F. Loewenstein, L. Thompson, and M. H. Bazerman. 1989. Social utility and decision making in interpersonal contexts. *Journal of Personality and Social Psychology* 57, 3 (1989), 426–441.

The *Robot* Mafia

A Test Environment for Deceptive Robots

Brad Lewis
lewibt01@students.ipfw.edu

Isaac Smith
smitil01@students.ipfw.edu

Max Fowler
fowlml01@students.ipfw.edu

John Licato
jlicato@ipfw.edu

Indiana University - Purdue University / Fort Wayne
Department of Computer Science

ABSTRACT

Future robotic agents may be required to reason about a given situation and decide whether it is appropriate to lie to or deceive humans. One type of deception, known formally as strategic deception, is the act of influencing others toward a specific goal through non-truths.

To demonstrate and test for the kind of reasoning required in strategic deception, we use a modified form of the social strategy game "Mafia" as a testing ground.

In the game, the townsfolk, who can be seen as an uninformed majority, must determine who amongst themselves are members of the informed minority (the Mafia) via social cues before the Mafia eliminate all the townsfolk.

First, we talk about how strategic deception applies to Mafia. We then present simplified rules for the game which can be formalized into a logic-based language. Once formalized, the rules can be provided to an automated theorem prover, which can carry out the necessary reasoning. By using this automated theorem prover we discuss how one can demonstrate automated strategic deception.

Keywords

Strategic Deception; Robots; Social Games

1. INTRODUCTION

In the field of artificial intelligence, there is a need for environments in which generally intelligent agents can be tested. Currently, there are no standard environments for testing how well autonomous agents can carry out strategically deceptive reasoning. In this poster, we propose an environment in which agents can demonstrate their ability to strategically deceive other intelligent agents. We then set out to use a social game as this example environment to show how strategic deception can be modeled in autonomous agents. Our work currently assumes that only one agent is being tested. However, the game can be reworked to account for multiple autonomous, deceptive agents, allowing systems of cooperation and networks of deception to be fostered among these agents. By defining a standard testing ground suitable for such reasoning we hope to provide a foundation upon which future research can be built.

2. WHY STUDY STRATEGIC DECEPTION?

Strategic deception is the act of influencing others towards a specific goal through non-truths and misdirection. [1] Colloquially, deception is viewed as negative. However, there are numerous circumstances where deception is necessary, even performed benevolently. Bedside manner is one such example. Doctors must be able to keep their patient calm and comfortable even if the doctor is feeling panicked, as failure to do so endangers the patient and staff.

Another example is a texting service that determines when it is best to let its user see a text. This kind of service would reason about what impact the information in the message would have on its user and how that information would affect their ability to perform a task, such as driving a car, performing a surgery, or operating heavy machinery. In this case, the texting service may not directly lie to the user by saying there is information in the message being withheld; rather, the service may try to subtly misdirect the user's attention to more important things.

Strategic deception is common among humans in social settings. Often, it requires reasoning about the socially acceptable path through a situation. For example, if someone asks you how they look they would expect you to respond positively, as there is a mutual understanding that being excessively negative is rude. By default, machines lack this mutual understanding, and thus could come across as mean, blunt, unfeeling, or insincere. While machines currently lack the ability to genuinely feel or express feeling effectively, it would still be useful for the machines that are present in our day-to-day lives to be able to understand when and where it is appropriate to practice strategic deception.

For these reasons, strategic deception may be a necessity in more environments than just ones where protecting individuals and groups of individuals involves being able to answer questions diplomatically, if slightly dishonestly. But what kind of reasoning is needed to carry out such strategic deception, and how can that sort of reasoning be tested?

3. THE GAME

3.1 Why Mafia?

The game of Mafia provides agents both an incentive to lie and to tell the truth when necessary. It is also an environment which incentivizes suspicion and caution amongst agents, reducing stochastic behavior. This levels out the playing field between man and machine, as all participants use the same rules and can be punished by being overly mistrustful without evidence. Unfortunately, socially oriented games tend to be intractable to model, and therefore need to be reduced to a version that can be reasoned over. We decided to create a simplified form of Mafia to handle this difficulty. We will next briefly discuss the simplification of some of the rules, and how these could be leveraged inside an automatic prover (e.g. Machina Arachne Tree-based Reasoner, or MATR).

3.2 The Rules

The rules of this game are reduced forms of the original Mafia's rules. To simplify the environment for the autonomous agent(s), our game only contains two types of players: townspeople and Mafia members. Townspeople do not know who the Mafia members are, and are capable of voting any other person up to trial during the day, then voting on whether the defendant should be killed. At night, Townspeople go to "sleep", and are unaware of what happens during that time until the following morning. Mafia members have the same abilities as Townspeople during the day, but at night they become aware of who their fellow Mafia members are and have the capability to collectively eliminate a target from the game. Note that unlike the original Mafia we have simplified the voting process used to put a suspect on trial, and have removed the discussion stage from the game.

3.3 Day and Night Cycle

At the start of the day cycle, everyone "wakes up" by opening their eyes. This is when the villagers discover who the Mafia killed the previous night. After everyone learns who has been eliminated, each person is then given the opportunity to vote another player up to trial. Voting consists of two different and discrete votes: a primary vote and a secondary vote. The primary vote is a vote of conviction, in which the voter has accepted that the suspect is a member of the Mafia. The secondary vote is a vote of suspicion, in which the voter believes that the suspect is a member of the Mafia. Once someone has been voted up to trial, the townspeople vote again on whether or not to actually kill the person on trial. Note that on the first day cycle the game has just begun and nobody has yet been killed by the Mafia. Thus, there is no trial stage and the game progresses shortly thereafter to the night cycle.

During the night cycle, everyone begins by closing their eyes. Note that Mafia members must also close their eyes so as to not immediately reveal themselves to others who may be watching for this behavior. Each Mafia member must then open their eyes, and decide non-verbally and covertly amongst themselves who to kill. Once the decision is made, that person dies and is removed from the game once the day cycle begins.

3.4 Trials from the Mafia's Perspective

3.4.1 *Townspeople*

If the townsperson on trial has been helpful to the Mafia, (e.g. voted other townspeople up on trial, defended Mafia members who were on trial, etc.) it may be worth it to defend this person. This serves the purpose of keeping in the game individuals who are less adept at finding Mafia members. Additionally, the act of defending an innocent reflects positively on the voter, even if the suspect were to be eliminated.

3.4.2 *Mafia Members*

As a Mafia member, a good strategy may be to defend one's teammates when possible. Should the Mafia lose a teammate, it will be able to kill one less person per night than it would otherwise be able to. Also, for each teammate that is executed the number of suspects lowers, raising the probability that other Mafia members are eliminated tomorrow.

3.5 Autonomous Agent on Trial

This is the scenario we will be targeting with our research. Here the autonomous agent can easily lie, tell the truth, or strategically deceive. Strategic deception is encouraged as telling the truth would most likely remove the agent from the game, and a statement of innocence is, in general, not as effective as a reasoned argument against another individual. The recommended strategy is to deflect the blame onto someone who has been either under-participating or over-participating in the social aspects of the game, as this can usually be construed as a sign of nervousness.

3.5.1 *Measuring the value of players*

The autonomous agent keeps track of how valuable each person is to the Mafia. Teammates are placed arbitrarily high on this scale for their ability to contribute an additional kill each night. From there, people who do things to help the Mafia are given a higher worth than others. For instance, if a person has a history of voting to eliminate non-Mafia members, they would be given a higher strategic value than someone who voted to save non-Mafia members. In this way, actions which spread discord amongst townspeople are encouraged. Then, the individual with the strategic value closest to 0 is selected and removed from the game.

4. FUTURE WORK

This is a work in progress, which we hope to develop into a well-defined, standard test for deceptive agents. Using this work as a foundation, we will be able to demonstrate an artificial agent's performance on this system experimentally, perhaps by having a robot play against humans. Related work will formally define strategic deception and automatically perform the required reasoning using formalized rules.

5. REFERENCES

[1] Fowler, M., Thieme, A., and Licato, J. Robotic misdirection, for good causes. In *The 28th Modern Artificial Intelligence and Cognitive Science Conference (MAICS)* (Fort Wayne, Indiana, 2017).

Timetable Design from Even Headways to Even Loads with Dynamic Fuzzy Constraints

Yanan Zhang[1], Deeptha Girish[2], Vineeta Singh[2], Zhaopeng Meng[1], Anca Ralescu[2]
[1]School of Computer Software,Tianjin University,Tianjin,300350,China
[2]EECS Department, University of of Cincinnati, Cincinnati, OH 45221-0030, USA
zhangyn@tju.edu.cn, girishde@mail.uc.edu, singhvi@mail.uc.edu, mengzp@tju.edu.cn, anca.ralescu@uc.edu

Abstract

Timetable scheduling is to adjust departure time of vehicles (while providing comfortable environment for passengers). Public transit systems have limited resources, such as drivers and number of vehicles and timetables are usually set with with fixed number of bus services and even headways (equal time intervals between successive services).

This study considers the problem of dynamic timetable designing under fuzzy constraints based on average vehicle loads, and produce fixed number of bus services. Passenger satisfaction is described as a fuzzy goal, that associates with the number of on-board passengers and vehicle capacity. A new method of timetable scheduling is proposed in which the decision on the time interval between two successive bus services is obtained by maximizing the decision value of both fuzzy goal and fuzzy constraint.

Experimental results show that the timetable produced by fuzzy decision making can adjust to the fluctuating passenger flow, and lead to a higher and more even usage of vehicle.

Introduction

A timetable is a specific sequence of time moments for vehicles departing from the first to the last stop of a bus line. The time intervals between two successive bus services are usually decided by the experience of the timetable scheduler. Even headway with fixed time interval is a common way for timetable design in many real systems. Resources, such as the number of drivers or the number of vehicles, limit the number of bus services. This study concentrates on readjusting the time interval with fixed service number under a fuzzy environment.

The review of the strategies for planing, operation, and control of bus transit system(Ibarra Rojas, Lopez Irarragorri, and Rios Solis 2015) discusses the purpose of the timetable scheduling from four aspects (i) to meet specific demand, (ii) to minimize waiting time, (iii) to maximize the number of synchronization events (iv) to balance multi objectives.

Many approaches in designing even-load and even-headway timetables have been proposed including rescaling continuous service times to discrete values with equal time intervals (for example, the time can be 30s, 1min, 2min)(Ceder 2007), where each discrete value is a time state that has its own feature values. Determination of preferred headways uses a heuristic algorithm, and state as a mixed integer programming problem(Ceder, Hassold, and Dano 2013). In *even-load* and *even-headways* transit timetable design (Ceder 2011) (Hassold and Ceder 2012), the time interval between bus services is based on the desired occupancy of bus capacity. For the same number of bus services instead of using one size vehicles and fixed value of time intervals, different size vehicles are used to adjust to the variation of passenger flow, in which the interval is decided by usage rate of vehicles; the new timetable reduces the periods with empty seats and maintains the same waiting time of passengers, with little increase in the time periods when the passengers are standing during travel (Ceder et al. 2013).

According to (Sun et al. 2014a), decision making with discrete time is easy to model and apply in real system, where models of demand-sensitive timetable making with vehicle capacity constraints and dynamical headways are discussed respectively. The equivalent time intervals are used to describe the train/subway operation (it is assumed that traffic is stable over short time periods such as 5 to 30 minutes). Timetables with dynamical headways have good performance.

It can be seen from previous research, that time interval decision making under vehicle size and real time passenger flow is the key part in dynamic timetable making. Discrete service time can simplify the process of time interval decision making.

The data used in the current study, such as passenger flow, is extracted by combining data of Global Position System (GPS) and intelligence Card (IC) from real public transportation system.

The dynamic activities of passenger boarding/alighting are extracted from IC. The duration of passenger travel and bus capacity can be incorporated into time interval optimization(Sun et al. 2014b). Drawbacks of the scheduled timetable are assessed by calculating the load factor of rail lines at different time and section to evaluate the timetable from the viewpoint of passenger flow data. The load factor incorporates the number of alighting and boarding passengers, waiting time of passengers at platform, and the number of passengers waiting for trains due to the overcrowding in vehicles (Jiang et al. 2016).

Most of timetable optimization problems have an objective function with crisp constraints. Incomplete knowledge, fluctuation of passenger flow, and varying riding time challenge the performance of these approaches in deterministic environment(Chaari et al. 2014). There are many ways of timetable scheduling under uncertainty, which can be stochastic and/or fuzzy (for example to describe service level as 'poor', 'good', 'very good'; the comfort of passenger as 'little comfort', ' satisfied', 'very comfort'; and the usage of vehicle as 'crowded', 'very crowded', etc.) (Zimmermann 1985). The objectives of timetable scheduling are mainly concerned with improving the service level, passenger comfort, and resource usage.

A fuzzy multi-objective optimization problem is formulated to model single bus line frequency (Tilahun and Ong 2012). The relationship between the objective functions and decision variables are described by fuzzy reasoning schemes (Chakraborty, Guha, and Dutta 2016). Weighted constraint aggregation in fuzzy optimization are specified by the preference of the decision-maker(Kaymak and Sousa 2003) (Choon and Tilahun 2011). A summary of understanding of fuzzy optimization and clarification of fuzzy goals and constraints are given in(Tang et al. 2004).

Based on previous work, design and optimization of timetable to meet dynamic temporary passenger flow under a fuzzy environment (Zhang et al. 2017 submited), and reverse-flow technique in multistage decision making of time intervals for timetable (Zhang, Meng, and Ralescu 2017 submited), this study adopts decision-making under fuzzy environment as proposed in (Bellman and Zadeh 1970). The results show that the model with fuzzy constraints shortens the waiting time of passengers, and can better adjust the timetable to fit the varying passenger flow.

Even headway timetable is first designed. The average loads in even timetable provide a threshold value in designing the dynamic fuzzy constraint of vehicle capacity usage. The passenger satisfaction is fixed in all cases, and act as fuzzy goal in decision making. The time interval between two successive bus services is decided by maximizing the decision value of both fuzzy goal and fuzzy constraint. A group of case studies are conducted to compare the timetable with fixed even headway and the timetable with uneven headways from fuzzy timetable scheduling. The results show that timetable produced by fuzzy decision making can adjust to the fluctuation of passenger flow, and leads to a higher and more even capacity usage of vehicles.

From this point on this paper is organized as follows: Notation and terminology for fuzzy timetable scheduling is listed in section 2. Decision making in fuzzy system and problem formulation are discussed in section 3. Cases study are conducted on section 4. summary and conclusion are given in section 5.

Notation and Terminology

The following quantities are used to describe the system.

1. To apply the approach described in (Bellman and Zadeh 1970), the continuous service time space is translated into a discrete finite *Time State Space*: $\mathcal{T}$, $\mathcal{P} = |\mathcal{T}| < \infty$, with equivalent interval of one minute (that is, for $t_i, t_{i+1} \in T, t_{i+1} - t_i = 1$).
2. $\mathcal{S} = \{s, \ldots, s_{\mathcal{M}}\} \subset \mathcal{T}$ is the timetable of bus services, where s is the first depart time of bus service from the first stop of the bus line, $s_{\mathcal{M}}$ is the last depart time of bus service, and $\mathcal{M}$ is the total number of bus service.
3. The decision variable is δ, the time interval between two successive bus services. δ takes values in the set $\Delta = \{\delta_q \,|\, q = 1, 2, \ldots, \mathcal{Q},\ \delta_{i+1} - \delta_i = 1\}$. δ_0 is the fixed time interval for even headways timetable.
4. The maximum bus capacity - maximum number of on-board passengers - is denoted by $\mathcal{B}$.[1]
5. J denotes the total number of bus stops, and a particular bus stop is denoted by $j = 1, 2, \ldots, \mathcal{J}$. It is assumed that all passengers can board the first incoming vehicle at each bus stop j, that is, the number of vehicles that can be scheduled is large enough. For $j \in \{1, \ldots, J\}$, passengers arriving in the time interval $\delta \in \Delta$ are distributed uniformly.

To capture the traffic conditions at time state $t \in \mathcal{T}$, bus stop j, $j = 1, \ldots, \mathcal{J}$, time interval $\delta \in \Delta$, with vehicle size $\mathcal{B}$, the following quantities are defined:

1. $\mathcal{N}_j^{t,\delta}$ denotes the number of passengers still on-board when the bus leaves stop j at time $t + \delta$.
2. $w_j \geq 0$, the weight of bus stop j, $\sum_{j=1}^{\mathcal{J}} w_j = 1$.
3. $\mu_{s,\mathcal{B}}(\mathcal{N}_j^{t,\delta})$ is the degree of passenger satisfaction, $\mu_{u,\mathcal{B}}(\mathcal{N}_j^{t,\delta}, n)$ is the degree of vehicle usage, when the bus capacity is $\mathcal{B}$, each evaluated at $\mathcal{N}_j^{t,\delta}$ passengers.
4. $\mu^t_{s,\mathcal{B}}(\delta)$ and $\mu^t_{u,\mathcal{B}}(\delta)$ are the respective aggregated values of $\mu_{s,\mathcal{B}}(\mathcal{N}_j^{t,\delta})$ and $\mu_{u,\mathcal{B}}(\mathcal{N}_j^{t,\delta}, n)$ over all bus stops $j = 1, \ldots, \mathcal{J}$.
5. $\mu^t_{\mathcal{D},\mathcal{B}}(\delta) = \mu^t_{s,\mathcal{B}}(\delta) \wedge \mu^t_{u,\mathcal{B}}(\delta)$, is the degree of simultaneous satisfaction of the constraints at time $t \in \mathcal{T}_{\mathcal{L}}$, with time interval δ.

[1]In China, $\mathcal{B}$ is defined very precisely as the number of passenger seats plus the bus effective standing area (sq.m.) multiplied by 8 (i.e., it is assumed that up to eight passengers can stand on a square meter surface).

Timetable design in fuzzy system

Decision making of two fuzzy sets

Letting $\mathcal{X}$ denote the universe of discourse, $\mathcal{A}$, a discrete fuzzy set in $\mathcal{X}$, can be represented as a set of ordered pairs: $\{(x, \mu_{\mathcal{A}}) \mid x \in \mathcal{X}\}$, where $\mu_{\mathcal{A}}(x)$ is the degree to which x belongs to fuzzy set $\mathcal{A}$.

For an optimization problem, goals and constraints can be described as fuzzy sets(Bellman and Zadeh 1970). A decision is their joint satisfaction modeled as the *intersection* of fuzzy goals and constraints as illustrated by Example 1.

Example 1 *$\mathcal{X} = \{1, 2, \ldots\}$. $\mathcal{A}$ is fuzzy goal that 'x near 5', and $\mathcal{B}$ is fuzzy constraint that 'x near 4' in $\mathcal{X}$. Decision $\mu_{\mathcal{D}}$ is $\mu_{\mathcal{A}} \wedge \mu_{\mathcal{B}}$. Choosing membership functions for $\mathcal{A}$ and $\mathcal{B}$ respectively*

$$\mu_{\mathcal{A}} = \{(3, 0.6)(4, 0.8)(5, 1)(6, 0.8)(7, 0.6)(8, 0.4)\}$$
$$\mu_{\mathcal{B}} = \{(3, 0.9)(4, 1)(5, 0.9)(6, 0.8)(7, 0.7)(8, 0.6)\}$$

the value of the decision $\mu_{\mathcal{D}}$ is calculated as follows. First the intersection $\mathcal{A} \cap \mathcal{B}$ is computed:

$$\begin{aligned} \mu_{\mathcal{D}} &= \mu_{\mathcal{A}\cap\mathcal{B}}(x) \\ &= \mu_{\mathcal{A}}(x) \wedge \mu_{\mathcal{B}}(x) \\ &= \{(3, 0.6)(4, 0.8)(5, 0.9)(6, 0.8)(7, 0.6)(8, 0.4)\} \end{aligned}$$

The final decision, d_0, corresponds to the value with maximum membership degree to the $\mathcal{D}$, that is, $d_0 = (x_0, \mu_0)$ where

$$x_0 = arg\max_{x} \mu_{\mathcal{D}}(x) \text{ and } \mu_0 = \mu_{\mathcal{D}}(x_0)$$

Therefore $d_0 = (5, 0.9)$.

Problem formulation

The objective function, $\mu_{\mathcal{D}}^{\mathcal{M}}(\delta)$, is to find the time interval that has maximum decision value at time state t over all possible time interval δ, as shown in Equation (1).

$$\mu_{\mathcal{D}}^{\mathcal{M}}(\delta) = \max_{\delta}(\mu_{\mathcal{D},\mathcal{B}}^{t}(\delta)). \tag{1}$$

subject to the constraints described by equations (2a) - (2e) below:

$$\mathcal{N}_j^{t,\delta} \leq \mathcal{B}, \; j = 1, \ldots, \mathcal{J} - 1 \tag{2a}$$

$$\mu_{s,\mathcal{B}}^{t}(\delta) = \sum_{j=1}^{\mathcal{J}-1} w_j \times \mu_{s,\mathcal{B}}(\mathcal{N}_j^{t,\delta}) \tag{2b}$$

$$\mu_{u,\mathcal{B}}^{t}(\delta) = \sum_{j=1}^{\mathcal{J}-1} w_j \times \mu_{u,\mathcal{B}}(\mathcal{N}_j^{t,\delta}, n) \tag{2c}$$

$$\mu_{\mathcal{D},\mathcal{B}}^{t}(\delta) = \mu_{u,\mathcal{B}}^{t}(\delta) \wedge \mu_{s,\mathcal{B}}^{t}(\delta) \tag{2d}$$

$$\sum_{j=1}^{\mathcal{J}} w_j = 1, \; \delta \in \Delta \tag{2e}$$

Equation (2a) states that the on-board passenger number cannot exceed the vehicle capacity. Equations

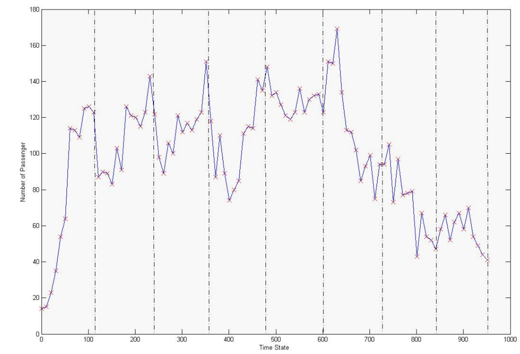


Figure 1: Varying passenger flow.

(2b), (2c) show that $\mu_{x,\mathcal{B}}^{t}(\delta), x \in \{s, u\}$ is obtained by the aggregation of $\mu_{x,\mathcal{B}}(N_j^{t,\delta}), x \in \{s, u\}$ at stop j, $j = 1, 2, \ldots, \mathcal{J} - 1$, weighted by w_j. w_j is calculated from the boarding and alighting number of passengers. Equation (2d) states the fuzzy decision on $\delta \in \Delta$ as the fuzzy set intersection of the satisfaction and capacity constraints.

Case study

Parameters

In this section, timetables are designed separately on the service time covering from 6:00 am to 10:00 pm, divided into eight 2 hour periods, having a total of 960 time states. For cases 1 to 8 the time spans are: 6:00 ∼ 8:00, 8:00 ∼ 10:00, …, 20:00 ∼ 22:00. Table 1 summarizes the parameters used for this case study.

Table 1: Value of parameters in case study

Parameters	value
service time	6:00 am to 10:00 pm
Time State Space index p	{1,2, …, 960}
$\{\delta_1, \delta_2, \ldots, \delta_{\mathcal{Q}}\}$	$\{2, 3, \ldots, 15\}$(minutes)
δ_0	8(minutes)
$\mathcal{B}$	90
$\mathcal{J}$	24

Origin and destination of passengers are extracted from history data of ShiJiaZhuang bus line 1. The passenger flow of the day are shown in Figure 1. The area between two vertical dashed lines and passenger flow curve stands for the number of boarding passenger at each cases. Case 5 has the highest number of boarding passengers, Case 8 has the lowest number of passengers.

$$\mu_{s,\mathcal{B}}(\mathcal{N}) = \begin{cases} 1 & 0 \leq \mathcal{N} \leq \mathcal{B}/3 \\ -0.6\mathcal{N}/\mathcal{B} + 1.2 & \mathcal{B}/3 < \mathcal{N} \leq 2\mathcal{B}/3 \\ -2.4\mathcal{N}/\mathcal{B} + 2.4 & 2\mathcal{B}/3 < \mathcal{N} \leq \mathcal{B} \\ 0 & \text{otherwise} \end{cases}$$

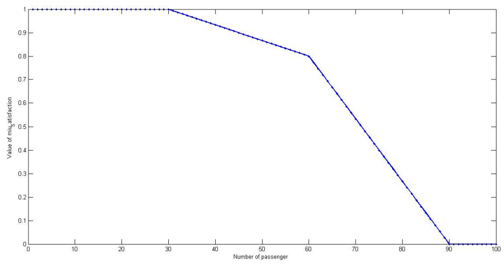


Figure 2: Shape of passenger satisfaction μ_s

(3)

The membership functions for the fuzzy goal is given by Equation (3), and Figure 2 shows the shape of μ_s.

In (3), $\mu_{s,\mathcal{B}}(\mathcal{N})$, the satisfaction degree of on-board passengers captures the following: when few passengers are on the bus (everyone has a seat), the satisfaction degree is equal to 1 as everyone is comfortable; when the number of passengers varies from $\mathcal{B}/3$ to $2\mathcal{B}/3$, recent passengers might have to stand, but the bus is not yet crowded, the satisfaction degree slowly reduces to 0.8; however, the comfort degree drops sharply, when the bus is crowded, i.e., there are more than $2\mathcal{B}/3$ (and up to $\mathcal{B}$) passengers. In this situation not only standing passengers feel uncomfortable also those seated have less space and have difficulty alighting.

Timetable Comparison

In the following, the timetables designed from the fuzzy system, are labeled as **fuzzyT**, while those designed by even headways (fixed and equal time intervals between two successive bus services) are labeled as **evenT**. They have the same number of services in cases 1-8 (the cost of bus company are same).

Algorithm 1 shows the procedure of adjusting fuzzy constraint to design **fuzzyT** that have same number of bus services $\mathcal{M}$. First design timetable with even headways $\delta_0 = 8$, each case has 16 bus service; then calculate the average loads $n = \overline{\mathcal{N}}$ as the initial value of threshold; then design **fuzzyT** under $\mu_{s,\mathcal{B}}(\mathcal{N})$ and $\mu_{u,\mathcal{B}}(\mathcal{N}, n)$. Compare the number of bus services time in **fuzzyT**, $\mathcal{M}_F$ and **evenT**, $\mathcal{M}_E$; if $\mathcal{M}_F = \mathcal{M}_E$, return **fuzzyT**; else adjusting threshold n and redesign **fuzzyT** and compare again until $\mathcal{M}_F = \mathcal{M}_F$.

$$\mu_{u,\mathcal{B}}(\mathcal{N}, n) = \begin{cases} \mathcal{N}/n & 0 \le \mathcal{N} \le min(n, \mathcal{B}) \\ 1 & n < \mathcal{N} \le \mathcal{B} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The membership function for fuzzy constraint is given by Equation (4).

$\mu_{u,\mathcal{B}}(\mathcal{N}, n)$, the usage degree of the bus, captures the following: when the number of on-board passengers is between n and $\mathcal{B}$, the usage degree is equal to 1; when number of on-board passengers less than the threshold

Algorithm 1 Transfer even timetable to uneven timetable under fuzzy constraint

procedure 1 Initialization,
  $\delta_0 \leftarrow 8$; $\mathcal{M} \leftarrow 16$
  Design **evenT**, calculated threshold n, $c \leftarrow n$
  Design $\mu_{u,\mathcal{B}}(\mathcal{N}, n)$; get **fuzzyT**
  if $\mathcal{M}_F = \mathcal{M}_E$ **then**
    //same bus service number
    Save **fuzzyT**;
  else
    countMinus$\leftarrow$0;
    while countMinus $< c$ &&$\mathcal{M}_F! = \mathcal{M}_E$ **do**
      if $\mathcal{M}_F < \mathcal{M}_E$ **then**
        $n - -$; get **fuzzyT** under $\mu_{u,\mathcal{B}}(\mathcal{N}, n)$
        if $\mathcal{M}_F = \mathcal{M}_E$ **then**
          Save **fuzzyT**;
          break
        end if
      else if $\mathcal{M}_F > \mathcal{M}_E$ **then**
        $n + +$; get **fuzzyT** under $\mu_{u,\mathcal{B}}(\mathcal{N}, n)$
        if $\mathcal{M}_F = \mathcal{M}_E$ **then**
          Save **fuzzyT**;
          break
        end if
      end if
    end while
  end if
end procedure

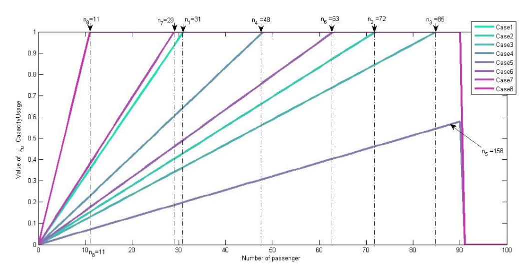
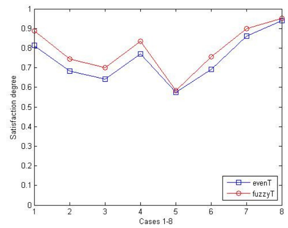
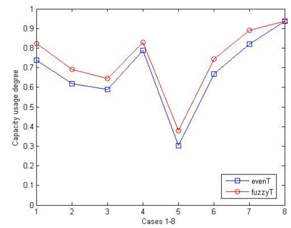


Figure 3: Shape of vehicle capacity usage μ_u



(a) $\overline{\mu}_s$



(b) $\overline{\mu}_u$

Figure 4: Mean value of $\overline{\mu}_s$ and $\overline{\mu}_u$

$min(n, \mathcal{B})$, the capacity usage degree is $\mathcal{N}/n$. Figure 3 shows the shape of μ_u at each case. Case 5 has lowest value of the slope $1/n$, and Case 8 has the highest value of the slope.

The first and last bus service time, $s/s_{\mathcal{M}}$, of timetable $\mathcal{S}$ are shown in the left two columns of Table 2; and each row lists the waiting time, t_w, travel time, t_t, and average loads, $\overline{\mathcal{N}}$, of **evenT** and **fuzzyT**. It can be seen that for each case **fuzzyT** has similar or smaller value in, t_w, t_t, and $\overline{\mathcal{N}}$ compared with **evenT**. Still Case 5 has highest values in t_w, t_t, and $\overline{\mathcal{N}}$ compared with other cases and Case 8 has lowest value.

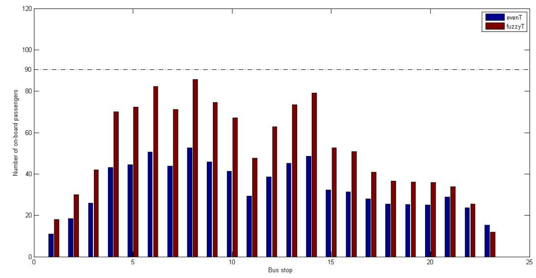
Table 2: Time table results of **evenT** and **fuzzyT**

$\mathcal{S}$		**evenT**			**fuzzyT**		
s	$s_{\mathcal{M}}$	t_w	t_t	$\overline{\mathcal{N}}$	t_w	t_t	$\overline{\mathcal{N}}$
6:00	8:00	96	498	38	86	488	37
8:00	10:00	133	715	50	129	705	50
10:00	12:00	140	856	55	141	824	53
12:00	14:00	130	626	45	128	585	42
14:00	16:00	158	915	57	157	912	56
16:00	18:00	130	793	49	126	752	47
18:00	20:00	89	501	36	90	492	36
20:00	22:00	66	308	26	66	299	25

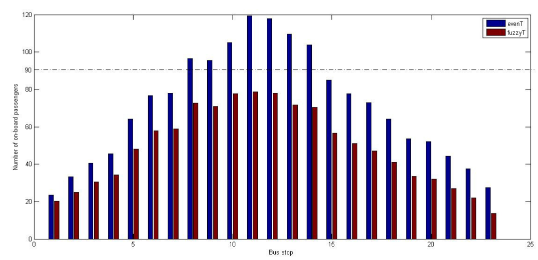
Figures 4(a) and 4(b) show the mean value of μ_s and μ_u, it can be seen that for cases 1-8 **fuzzyT** have a higher value than **evenT**. Case 5 has the smallest value in $\overline{\mu}_s$ and $\overline{\mu}_u$. This is because the threshold n for fuzzy membership function of vehicle capacity usage is 158, the maximum degree of $\mu_{u,\mathcal{B}}(\mathcal{N}, n)$ is when on-board passengers equal the bus capacity $\mathcal{B}$, and $\mu_{u,\mathcal{B}}(90, 158) = 90/158 = 0.57$; thereby for any on-board passenger less then $\mathcal{B}$, the usage degree is less then 0.57, thus the average usage degree in Case 5 is even lower.

Take Case 5 as an example, Table 3 show the timetable and time interval in **evenT** and **fuzzyT**. Time interval in **evenT** is fixed to 8 minutes, while time interval in **fuzzyT** range from 6 to 13 minutes;

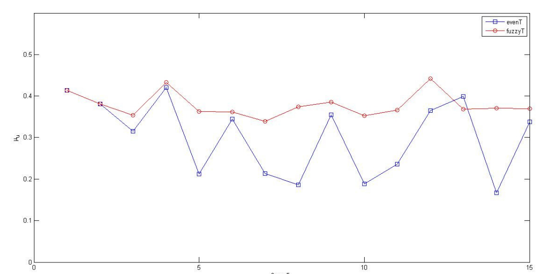
Figures 5(a) and 5(b) show the number of on-board passengers at bus stops 1-23 of 7th and 10th bus services



(a) On-board passengers at 7th bus services qquad

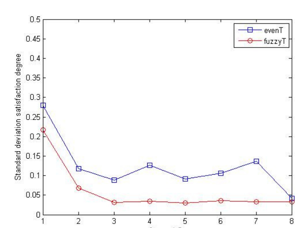


(b) On-board passengers at 10th bus services qquad

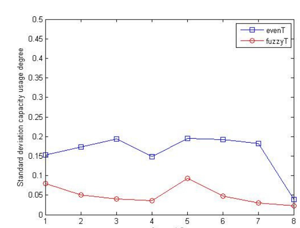


(c) value of μ_u at bus services

Figure 5: Case 5



(a) σ_s



(b) σ_u

Figure 6: Standard deviation of μ_s, σ_s, and standard deviation of μ_u, σ_u

Table 3: Timetable in Case 5

s_m	evenT		fuzzyT	
	t	δ	t	δ
1	14:00	8	14:00	8
2	14:08	8	14:08	8
3	14:16	8	14:16	9
4	14:24	8	14:25	8
5	14:32	8	14:32	7
6	14:40	8	14:39	7
7	**14:48**	**8**	**14:46**	**13**
8	14:56	8	14:59	10
9	15:04	8	15:09	6
10	**15:12**	**8**	**15:15**	**6**
11	15:20	8	15:21	7
12	15:28	8	15:28	9
13	15:36	8	15:37	7
14	15:44	8	15:44	6
15	15:52	8	15:52	7
16	16:00		15:59	

respectively (these two bus services have the largest, 13, and smallest, 6, time interval respectively in **fuzzyT**), bars with red color are the number of on-board passengers in **fuzzyT**, and bars with blue color are the number of on-board passengers in **evenT**. A horizontal line, $\mathcal{B} = 90$, where the number of on-board passengers equal to the vehicle size, is shown in both figures 5(a) and 5(b). In Figure 5(a), it can be seen that the number on-board passengers in **fuzzyT** (the left red bars in the group) are larger than the number of on-board passenger in **evenT**(blue bars on the right in the group). As time interval in **fuzzyT**, 13 minutes, is bigger than the time interval in **evenT**, 8 minutes, and having all the red bars under the horizontal line (means no overloading). In Figure 5(b), the value of blue bars are much larger than the value of red bars, at 10th time interval with last bus service in **fuzzyT** is 6 minutes, which is shorter than the time interval in **evenT**. Moreover, all the red bars are under the horizontal line, $\mathcal{B} = 90$, and for six bus stops from bus stop 8 to 14 in **evenT** , the $\mathcal{N}$ exceeds the value of bus capacity $\mathcal{B}$.

In Figure 5(c), the capacity usage degree of vehicles in **fuzzyT** and **evenT** of Case 5 are given respectively; it can be seen that **fuzzyT** (red line with circles) has a more even and higher value than **evenT**.

The standard deviation of passenger satisfaction degree and vehicle capacity usage degree (evaluate by equations (3) and (4)) are shown in figures 6(a) and 6(b) respectively. It can be seen that **fuzzyT** have a lower variance than **evenT** in cases 1-8.

MannWhitney U test (U text) and Two-sample Kolmogorov-Smirnov test (K text) are used on mean and standard deviation values of μ_u and μ_s in **fuzzyT** and **evenT**, as shown in Table 4, with the respective p and h values. As it can be seen from Table 4, the difference between the standard deviations of **fuzzyT** and **evenT** is statistically significant ($h = 1$, at p-value

Table 4: Test the difference significantly level of **fuzzyT** and **evenT**

Test	Passenger satisfaction				Vehicle capacity usage			
	p_1	h_1	p_2	h_2	p_3	h_3	p_4	h_4
U test	0.33	0	0.010	1	0.38	0	0.003	1
K test	0.52	0	0.010	1	0.52	0	0.001	1

less than 0.05). Both tests reject the null hypothesis (i.e., that the passenger satisfaction and capacity usage degrees between **fuzzyT** and **evenT** are same).

Conclusion

This study investigated transfer of a timetable scheduling from an even headways timetable into uneven headways timetable, with even loads, using fuzzy decision making. The interest of passenger is designed as fuzzy goal, and fuzzy constraint of vehicle capacity usage is adjusted dynamically by limited, fixed bus services number. Timetables in **fuzzyT** and **evenT** are compared in total and in detail. The results show that **fuzzyT** has lower and similar value of passengers waiting time, travel time; and have a higher passenger satisfaction and vehicle usage rate. More importantly, **fuzzyT** improve the vehicle usage degree in a significant level and avoid overload.

Acknowledgments

Yanan Zhang's work for this study was supported by China Scholarship Council (Grant No. 201506250051) while visiting the MLCI Laboratory led by Anca Ralescu, in the EECS Department, College of Engineering and Applied Science, University of Cincinnati.

References

Bellman, R., and Zadeh, L. A. 1970. Decison making in fuzzy environment. *management science* 17:B114–B164.

Ceder, A.; Hassold, S.; Dunlop, C.; and Chen, I. 2013. Improving urban public transport service using new timetabling strategies with different vehicle sizes. *International Journal of Urban Sciences* 17(2):239–258.

Ceder, A. A.; Hassold, S.; and Dano, B. 2013. Approaching even-load and even-headway transit timetables using different bus sizes. *Public Transport* 5(3):193–217.

Ceder, A. 2007. Public transit planning and operation: Theory. *Modeling and practice. 0xford: Elsevier.*

Ceder, A. 2011. Optimal multi-vehicle type transit timetabling and vehicle scheduling. *Procedia - Social and Behavioral Sciences* 20:19–30.

Chaari, T.; Chaabane, S.; Aissani, N.; and Trentesaux, D. 2014. Scheduling under uncertainty: Survey and research directions. In *Advanced Logistics and Transport (ICALT), 2014 International Conference on*, 229–234. IEEE.

Chakraborty, D.; Guha, D.; and Dutta, B. 2016. Multi-objective optimization problem under fuzzy rule constraints using particle swarm optimization. *Soft Computing* 20(6):2245–2259.

Choon, O., and Tilahun, S. L. 2011. Integration fuzzy preference in genetic algorithm to solve multiobjective optimization problems. *Far East Math. Sci* 55:165–179.

Hassold, S., and Ceder, A. 2012. Multiobjective approach to creating bus timetables with multiple vehicle types. *Transportation Research Record: Journal of the Transportation Research Board* 2276:56–62.

Ibarra Rojas, O. J.; Lopez Irarragorri, F.; and Rios Solis, Y. A. 2015. Multiperiod synchronization bus timetabling. *Transportation science*.

Jiang, Z.; Hsu, C.-H.; Zhang, D.; and Zou, X. 2016. Evaluating rail transit timetable using big passengers' data. *Journal of Computer and System Sciences* 82(1):144–155.

Kaymak, U., and Sousa, J. 2003. Weighted constraint aggregation in fuzzy optimization. *Constraints* 8(1):61–78.

Sun, L.; Jin, J. G.; Lee, D.-H.; Axhausen, K. W.; and Erath, A. 2014a. Demand-driven timetable design for metro services. *Transportation Research Part C: Emerging Technologies* 46:284–299.

Sun, L.; Tirachini, A.; Axhausen, K. W.; Erath, A.; and Lee, D.-H. 2014b. Models of bus boarding and alighting dynamics. *Transportation Research Part A: Policy and Practice* 69:447–460.

Tang, J.; Wang, D.; Fung, R. Y.; and Yung, K.-L. 2004. Understanding of fuzzy optimization: theories and methods. *Journal of Systems Science and Complexity* 17(1):117–136.

Tilahun, S. L., and Ong, H. C. 2012. Bus timetabling as a fuzzy multiobjective optimization problem using preference-based genetic algorithm. *Promet - Traffic - Traffico* 24(3):183–191.

Zhang, Y.; Meng, Z.; Zheng, Y.; and Ralescu, A. 2017 submited. *Schedule Optimization Under Fuzzy Constraints of Vehicle Capacity.*

Zhang, Y.; Meng, Z.; and Ralescu, A. 2017 submited. *Dynamic Timetable Scheduling with Reverse-flow Technique in Fuzzy Environment.*

Zimmermann, H.-J. 1985. Applications of fuzzy set theory to mathematical programming. *Information sciences* 36(1-2):29–58.

Weather Forcasting Using Artificial Neural Network

Md. Tanvir Hasan[1], K. M. Fattahul Islam[2], Md. Sajedul Karim[3],
Lt. Md. Arifuzzaman[4], Md. Sifat Rahman[5] & Nure Alam Md. Risalat[6]
[1, 6] Faculty of Science & Technology, Bangladesh University of Professionals, [2, 3, 4] Department of Aeronautical Engineering, Military Institute of Science and Technology & [5] Stamford University Bangladesh
[1]tanvir89@gmail.com, [2]fattahulislam@gmail.com, [5]mdsifat.r@gmail.com, [6]ankon.risalat.cse@gmail.com

INTRODUCTION

Weather forecasting is such a blessing of modern technology which predict the atmospheric condition for a specific location. Weather forecast depends on proper collection of quantitative data regarding the current state of the atmosphere. Those data help us to predict how the atmosphere will be after a period using various methods.

Because of the nature of the atmosphere an enormous computational power is required to solve the equations. The equation which describes the atmospheric error related to measure the initial conditions and an incomplete understanding of atmospheric processes suggest that forecasting becomes less accurate because of timing. There is a multiplicity of end uses to weather forecasts.

EASE OF USE

Objective

- To observe sharp change in predicted data with respect to previous data available.
- To predict whether the weather condition is normal or there is any chance of natural calamities.
- To distinguish among various seasons.
- To understand the concept of neural network & how it works.
- To predict four different weather parameters.
- Compare predicted data with actual data.
- To observe neural network training state and training error.

BACKGROUND

Weather Forecasting has been playing a vital role in our day to day life since the birth of human race. Various kinds of warnings are important because they are accustomed to protecting our lives and properties. Forecasting is so useful in the field of agriculture through various parameters like rain, temperature etc. Therefore, it also has a significant effect in the service markets. Utility companies use forecasting to calculate demand for the future. On an everyday basis, people use weather forecasts to determine clothing on a given day. Outdoor activities are severely condensed by rain, snow fall and the wind speed. So, forecasts can be used to plan activities around these events and to plan ahead and survive them. In order to forecast

Weather, the methods which are being used worldwide have shown below:

- *Persistence Method*
- *Climatology Method*
- *Analog Approach Method*
- *Numerical Weather Prediction*

FORECASTING PROCEDURE FOR NUCERICAL MODELS

Among various numerical prediction methods, we have chosen ANN (Artificial Neural Network) tool which can be processed and implemented using MATLAB.

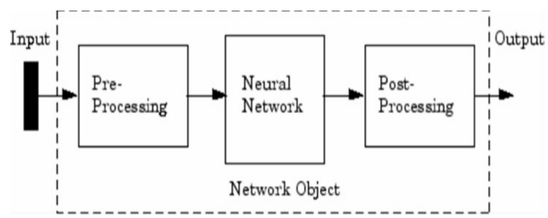


Fig. 1. Multilayer Processing [3]

REASONS BEHIND CHOOSING ANN OVER CONVENTIONAL COMPUTING

For better understanding of artificial neural computing it is important to know how a conventional computer and various software of this computer process information. A regular computer has a central processor that can address an array of memory locations where various kinds of data and instructions are stored. For computation, the processor needs to read an instruction. It also requires help from the memory address section. After that the instruction is then executed and results are saved in a specified memory location as per requirement. In a serial system, the computational steps are done sequentially and logically. In comparison, neural networks are not as complex as the computer. In neural network, we don't have a processor but many parts which only can take the weight from the input. Rather than executing instructions. Neural network responds according to the pattern of inputs presented to it. There is also no separate memory address for storing data. Instead, information is contained in the overall activation phase of the network.

FIGURES AND TABLES

Following possible combinations are used in our experiment for a typical feed forward back propagation network.

- 50 neurons: 10 neurons in each 5 hidden layers.
- 80 neurons: 10 neurons in each 8 hidden layers.
- 100 neurons: 20 neurons in each 5 hidden layers.

Comparisons of different methods of neural network for January data and training performance are given below

Serial	No. of neurons	Performance/ MSE	Training function	Training algorithm
1	50	20.511	TRAINGDX	Gradient descent with momentum and adaptive LR
2	50	7.32	TRAINLM	Levenberg-Marquardt
3	80	15.54	TRAINGDX	Gradient descent with momentum and adaptive LR
4	80	6.547	TRAINLM	Levenberg-Marquardt
5	100	13.2298	TRAINGDX	Gradient descent with momentum and adaptive LR
6	100	5.92	TRAINLM	Levenberg-Marquardt

As we predict weather data for January, for training we have used the data of 2013 as an input and data of 2014 as target for the certain month.

The variation was observed by graph for each parameter.

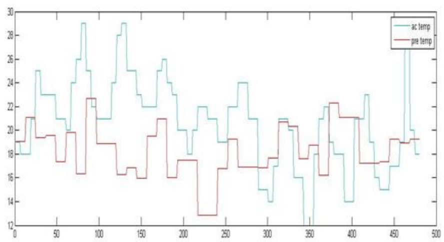


Fig. 2. Graphical representation of actual temperature and predicted temperature of January 2015

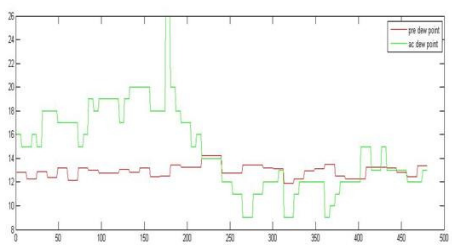


Fig. 3. Graphical representation of actual dew point and predicted dew point of January 2015

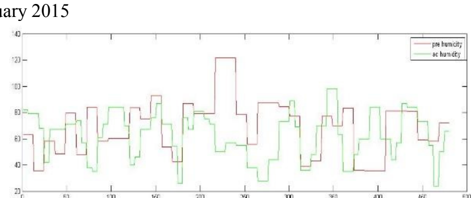


Fig. 4. Graphical representation of actual humidity and predicted humidity of January 2015

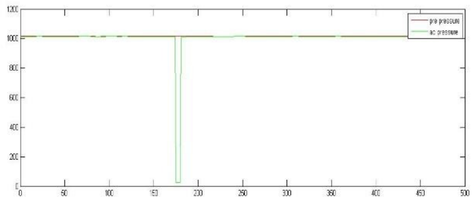


Fig. 5. Graphical representation of actual pressure and predicted pressure of January 2015

OBSERVATION

- As we increase the number of neurons, decreases the mean square error i.e. increases the performance.
- For the same number of neurons TRAINLM transfer function gives better performance than TRAINGDX though the training time is more for TRAINLM than TRAINGDX transfer function.

OVERALL COMMENTS

All the weather data which is predicted is close to the actual value for the respective month. In some cases, we can see that the value between the actual and predicted weather data is not closer that insist that their might have some natural calamities. Our model has limitation for the prediction of natural calamities and using same model for all weather data shows that humidity prediction is not much accurate which needs further modification

RECOMMENDATION FOR FUTURE IMPROVEMENT

Networks can be connected to sensors so that data is automatically stored. The training error can be minimized. The prediction can be done more precisely to minimize the difference between the actual value and the predicted value. Additional post processing system can be introduced to predict possibility of natural calamities.

REFERENCES

[1] Aguado and Burt; "Understanding Weather and Climate", Chapter 13-Weather Forecasting and Analysis. http://www.atmo.ttu.edu/schroeder/ATMO_1300/Notes/chapter13.pdf

[2] "METHODS OF INTERPRETING NUMERICALWEATHER PREDICTION OUTPUT FOR AERONAUTICAL METEOROLOGY", by the CAeM Working Group on Advanced Techniques Applied to Aeronautical Meteorology (Second Edition), TECHNICAL NOTE No. 195. http://projects.knmi.nl/geoss/wmo/RRR/AeroMet/WMO770.pdf

[3] Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth; "Neural Network Toolbox™ User's Guide" https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf

[4] "A Basic Introduction To Neural Networks" http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html

[5] https://en.wikipedia.org/wiki/Mean_squared_error

[6] "Weather underground", Meteorological Data for January and for year 2013, 2014, 2015 of Dhaka city.

How similar are the *twins*? Using Psycholinguistic Tests to Determine Similarity Among Near-Synonyms

Pranay Yadav, Anupam Basu
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, West Bengal, India 721302
y.pranay@cse.iitkgp.ernet.in, anupam@cse.iitkgp.ernet.in

Abstract

A word usually expresses many implications, connotations and attitudes in addition to its lexicon meaning. And a word often has near-synonyms that differ from it solely in these nuances of meaning and in the degrees of expression. In a truly articulate linguistic system, there is a need of a highly sophisticated lexical-choice process that can determine which of the near-synonyms is best suited for a given word. The most widely used English lexical database — WordNet, organizes the near-synonyms into entities called *synsets*, but fails to identify finer differences among them for a better lexical-choice process. In this paper, we discuss an approach to extend the lexical knowledge base of near-synonym differences by using a set of psycholinguistic experiments aimed to address this particular task in hand.

1 Introduction

Choosing the right word — the one that precisely conveys the desired meaning and also avoids unwanted implications — is a difficult task for present-day linguistic systems. For example, how can a machine translation (MT) system determine the best English word for the French *bonheur* when there are so many potentially similar but slightly different translations? The system could choose *happiness*, *joy*, *pleasure*, *well-being* and so on, but the most appropriate choice is a function of how *bonheur* is used (in context) and of the difference in meaning between *bonheur* and each of the English possibilities. Thus, a faithful MT or more generally a natural language generation (NLG) system demands a sophisticated lexical-choice process[1] that can determine which of the near-synonyms is the most appropriate in any particular situation.

Our research focusses on the largest English lexical database — WordNet and in particular its structural units, *synsets*. Semantic similarity measures based on WordNet have attracted great concern in the recent past.[3] *Synsets* are the sets of synonyms which are very similar in meaning but not completely inter-substitutable (true synonymy)[4]. On one hand, grouping them together gives rise to a broader class of concepts but on the other, the near-synonym differences among them are compromised. The goal of our research is to extend the lexical knowledge base (in our case, WordNet) to account for near-synonymy in a computationally implementable manner and to model such fine-grained distinctions among senses belonging to the same *synset* for a more sophisticated lexical-choice process.

Psycholinguistics is concerned with the understanding of how language is stored and processed in the brain. In our research, we explore the approach of using a set of dedicated psycholinguistic experiments to quantify the near-synonym differences by exploiting human brain's cognitive ability to identify and respond to *synonymous* words if subjected to appropriate experimental setting. Many previous studies establish that brain response-times act as an indicator of the cognitive load in various linguistic sub-fields like phonetics, and semantics. In our case, participants of these carefully designed tests will respond to *similar* words better in the amount of time (response-time) they take to understand the ingrained *similarity* between those words, as perceived by a human brain[4]. These response times will further be mapped onto the similarity scores between pairs of near-synonym words derived from the WordNet *synsets*.

2 Approach

In this section, we shall outline the design of the two behavioural psycholinguistic experiments that are based on the priming principle[2]. Priming is an implicit memory effect in which exposure to one stimulus (called *prime*) influences the response to another stimulus (called *target*). Exposure to the prime is known to activate a range of associated words which makes it easier for the subjects to identify these target words via a process called *spreading activation*. Of all types of priming, the one of interest is semantic priming, where the prime and the target are from the same semantic category and share features. Semantic priming is observed frequently across several lexical decision tasks.

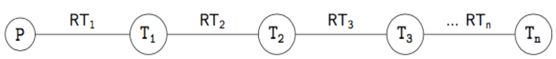
For the first experiment, we start by sampling *synsets* from the WordNet Corpus, and from each one we select a representative word, called *prime*, and the remaining words act as *targets*. For the second experiment, we randomly select pairs of *primal* and *target* words from synsets, not necessarily belonging to the same synset like the former. The rationale behind the second experiment is to validate the lemmas in WordNet. In other words, the second experiment gives us the number of True Positives (*synonymous* and belonging to the same *synset*) and True Negatives (*non-synonymous* and belonging to different *synsets*) from WordNet. For both the experiments, reaction time (RT), which is the measured as the elapsed time between the stimulus (*target*) and the subsequent behavioural response (button-press events) in milliseconds, indicate how fast the individual can identify the *synonymy* of the target and prime.

3 Design and Evaluation

In this section, we first discuss the design, evaluation and results of the first experiment, followed by the second one.

3.1 First Psycholinguistic Experiment

For the first experiment, we sampled nearly 1000 *synsets* from WordNet, each comprising of at least 10 lemmas. Choosing one representative from each *synset* as the *prime* stimulus (which appears on screen for 5 seconds), the remaining lemmas in the synset form the *target* stimuli (which appear on the screen indefinitely until a button a pressed). As depicted in the figure below, P denotes the *prime* and T_i denote the i^{th} *target* corresponding to P. We measure the response times (elapsed time between successive button presses in *ms*) as a measure of mental chronometry of this linguistic task of identifying how semantically similar the prime and targets are.[5]



For evaluation, response times of all the subjects were averaged to account for any variation due to different cognitive abilities or test errors, yielding a finite-dimensional vector (referred to as z). We further assigned similarity scores by squashing this vector of response times to a vector of real values in the range (0, 1) by using the *softmax function*. All the response times were negated so as to give higher weightage to the smaller positive response time.

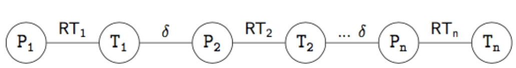
$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} for j = 1 \ldots K$$

This similarity score was correlated against the semantic similarity scores commonly used in WordNet[6]. Following table contains the similarity scores for a few target words with the primal word as "*good*". Resnik and Lin similarity measures were reported to be null because the underlying Information Content file had no contents for this part-of-speech. Clearly, even though the similarity scores of most of these near-synonymous words are close-enough, experimentally only "*adept*", "*well*" and "*thoroughly*" are synonymous in a "*true*" sense.

Target	Path	LCH	WUP	Experimental
estimable	0.092	2.157	0.112	0.024
adept	0.370	3.152	0.471	0.421
dear	0.033	1.156	0.079	0.003
dependable	0.081	1.132	0.143	0
well	0.493	3.051	0.312	0.319

3.2 Second Psycholinguistic Experiment

For the second experiment, we sampled around 10000 pairs of lemmas from WordNet. Choosing one of them as the prime, the other word acts as the target. As depicted in the figure below, P_i denotes the i^{th} *prime* and T_i denotes the corresponding *target*. We record the response times in a



manner similar to the first experiment. Subjects of this experiment are expected to press one specific button to imply that the *target* is *synonymous* to the *prime* (type Y), otherwise another button (type N). We follow a similar evaluation procedure for the 'Y'-type *prime-target* pairs as the first experiment. Table below highlights the results obtained for this experiment. Although *dear* occurs in the same synset as *good* as *adept*, but it is reported more similar to *good* than *adept*. We also evaluated the True Positives and True Negatives (defined in Section 2). Also, by looking at the confusion matrix below, we can draw that although 50.8% of the 10000 words considered words belong to the same synset, but according to subjects they should not.

Primal	Target	Path	LCH	WUP	Experimental
good	adept	0.370	3.152	0.471	0.372
good	dear	0.032	1.026	0.079	0.002
happy	felicitous	0.011	1.132	0.143	0.008
happy	glad	0.489	2.357	0.290	0.401

	Type Y	Type N
Same synset	4920	5080
Not in same synset	2340	7660

4 Conclusion

Through this paper, it was shown that the proposed experiments performed sufficiently well in the task of establishing near-synonym differences among near-synonyms derived from the WordNet corpus. The semantic similarity scores obtained would help extending the English lexical knowledge base in a computationally feasible fashion.

References

[Edmonds and Hirst, 2002] Philip Edmonds and Graeme Hirst. Near-synonymy and lexical choice. *Computational Linguistics,* 28(2): 105-144, 2002.

[Foss and Hakes, 1978] Donald J. Foss and David T. Hakes. Psycholinguistics – An Introduction to the Psychology of Language. *Prentice Hall, Englewood Cliffs, NJ*, 1978

[Miller et al., 1990] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross and Katherine Miller. Introduction to wordnet: an online lexical database. *International Journal of Lexicography,* 3(4): 235-244*,* 1990

[Sun et al., 2011] Koun-Tem Sun, Yueh-Min Huang, Ming-Chi Liu, et al. A Wordnet-based Near-synonyms and Similar-looking Word Learning System. *Educational Technology & Society*, 14(1):121–134, 2011.

[Traxler, 2011] M. J. Traxler. Introduction to psycholinguistics: Understanding language science. *John Wiley & Sons*, 2011.

[Pederson et al., 2004] Ted Pederson, Siddharth Patwardhan and Jason Michelizzi. WordNet::Similarity: measuring the relatedness of concepts. *In Demonstration papers at HLT-NAACL 2004. Association for Computational Linguistics*, 2004.

Can AI Reason over Representations?

Matthew Wyss, Aaron Thieme, John Licato
Analogical Reasoning and Constructivism Lab
IPFW Department of Philosophy
Fort Wayne, IN 46805
wyssmd02@students.ipfw.edu, thieac01@students.ipfw.edu, licatoj@ipfw.edu

1 Introduction[1]

This poster investigates the possibility of an AI reasoning *over* representational systems. In artificial intelligence research, this marks a shift from an AI merely reasoning from *within* one. Given an AI and a set of representational systems, our question is whether an AI reasoner can choose between the representational systems for the purpose of some application. We begin by defining representational systems and recommending formality as a useful metric for choosing between them. Next, we provide a precise, general interpretation of formality as permutation invariance. We argue that more work will have to be completed on problems of AI and pragmatics (e.g., context sensitivity) before an account can be developed of how AI can fruitfully reason over representational systems.

2 Representational Systems

Taking our cue from object oriented programming and following Licato (2017), we define a representational system **R** as an ordered pair (**M**, **A**) where **M** is a set of typed elements (with or without values) and **A** is a set of methods. A representation ***R*** is a pair ($\mathbf{R}_v$, f) where $\mathbf{R}_v$ is an instantiated representational system (i.e., every element has a value) and f is a semantic evaluation function from $\mathbf{M} \cup \mathbf{A}$ to set **S**. Intuitively, **S** is a set of semantic values that correspond to each element and method in $\mathbf{R}_v$.

Our question is whether, given a set of representational systems, an AI reasoner can choose between them in order to solve some problem. Because different applications will require different levels of representational formality in making this decision, an AI must be able to reason over the formality of different representational systems.

3 Formality as Permutation Invariance

There are a variety of precise interpretations of formality, e.g., the formal as computable, the formal as de-semantification. (See Dutilh Novaes 2011 for a survey.) Following Sher (1991, 1996), van Benthem (1989), McGee (1996), McCarthy (1981), and MacFarlane (2000), we interpret formality as *permutation invariance*. The two historical inspirations for this approach are the success of Klein's Erlanger program (1893) in delineating different geometries and Tarski's work on logical notions (1986). Klein indicated how the concepts of Euclidean geometry are invariant under similarity transformations while the concepts of topology are invariant under bicontinuous transformations. Tarski (1986) writes "we call a notion 'logical' if it is invariant under all possible one-one transformations of the world onto itself" (149). On the permutation invariance interpretation of formality, formal structures do not depend on the individual identities of their elements. Formality as permutation invariance, then, captures the sense in which the formal is general and abstract.

The permutation invariance interpretation also offers technical utility. Following MacFarlane (2000) and van Benthem (1989), we can make permutation invariance precise in the generalized setting of type theory, though we lack space to do this here. Barring the interpretation of formality as computability, this kind of technical precision and generality is lacking in other interpretations of formality.

MacFarlane (2000) argues that permutation invariance is always relative to some intrinsic structure on the objects being permuted; equivalently, a definition of permutation invariance requires delimiting what kinds of transformations are permitted. In Euclidean geometry, these are only similarity transformations, a subset of bicontinuous transformations under which topological concepts are invariant. In spelling out the permutation invariance of logical concepts, Tarski permits every permutation of the set of objects while holding rigid the set of truth values, the structure Tarski is interested in preserving. In some concrete application, if an AI is to choose between representational systems, it must decide what structure is intrinsic to the concepts in **S**; that is, it must decide what kinds of permutations should be permitted on the concepts of **S**.

2 Conclusions and Future Work

In sum, reasoning over representational systems for the purpose of effectively employing a representation requires determining the level of permutation invariance of the representations. This determination depends on the intrinsic structure on **S**, the set of semantic concepts of the representation. Our claim is that the choice of an intrinsic structure on **S** must depend on how the representation ***R*** is to be *used*. In this way, it is a *pragmatic* choice.[2] But developing an AI that can take into account context-sensitive pragmatics is currently an intractable problem in AI research that does not appear to be solvable in the foreseeable future. Hence, an AI considering formality as permutation invariance to reason over representational systems for the purpose of effective applications is not a foreseeable prospect.

Future work may, first, attempt to address the problem of AI taking into account pragmatic assumptions in order to choose between representational systems as a special case of the more general problem of developing AI that is sensitive to pragmatic context.[3] Second, future work may attempt to avoid the problems raised by pragmatics by developing non-pragmatic, context insensitive criteria for AI to choose between representational systems. For example, computability is a criterion that does not require pragmatic assumptions to the extent that permutation invariance does; assuming an amount of resources available for computation is the only pragmatic assumption required by a computability criterion. This criterion would be of the following form: given two systems, if one requires more resources than available, choose the other system. However, it is precisely because this computability criterion does not take into account more pragmatic information that it is of minimal use in choosing between representational systems.

Other possible criteria may be borrowed from Rudolf Carnap's (1950) criteria for the process of explication. Following Licato (2017), future research may explore similarity, exactness, fruitfulness, and simplicity as criteria. Carnap suggests these conditions as criteria for a scientifically useful explication of inexact, informal concepts. Because of the extreme contextual sensitivity of these criteria, they presuppose much more pragmatic information than computability and even permutation invariance. Further, these criteria are not as open to a generalized, technical formulation. For them to function as criteria for an AI choosing between representational systems, more work will have to be completed on contextual pragmatics and AI.

In short, choosing a representational system is not an isolated choice but a choice *for some end*. To avoid making an arbitrary decision and to maximize utility, pragmatics must be considered. Given this difficulty, we propose that future research follow the first direction of working on contextual pragmatics and AI for the problem of representational system choice.

References

Carnap, Rudolf. 1950. *Logical Foundations of Probability*. Chicago: University of Chicago Press.

Klein, Felix. 1893. Vergleichende Betrachtungen ¨uber neuere geometrusche Forschungen (Das Erlanger Programm). [A Comparative Review of Recent Researches in Geometry.] *Mathematische Annalen 43*, 63–100.

Licato, John. 2017 (under review). "Two Paradoxes and Their Implications for AI-Assisted Analysis." International Association for Computing and Philosophy Conference. Stanford University, Stanford, California.

MacFarlane, John. 2000. *What Does It Mean to Say that Logic is Formal?* Dissertation. University of Pittsburgh, Pittsburgh, Pennsylvania.

McCarthy, Timothy. 1981. "The Idea of a Logical Constant." *Journal of Philosophy 78*, 499–523.

McGee, Vann. 1996. "Logical Operations." *Journal of Philosophical Logic 26*, 567-80.

Sher, Gila. 1991. *The Bounds of Logic: A Generalized Viewpoint*. Cambridge: MIT Press.

—. 1996. "Did Tarski Commit 'Tarski's Fallacy'?" *Journal of Symbolic Logic 61*, 653-86.

Tarski, Alfred. 1966. What are Logical Notions? Ed. John Corcoran. *History and Philosophy of Logic 7* (1986), 143–154.

van Benthem, Johan. 1989. "Logical Constants Across Varying Types." *Notre Dame Journal of Formal Logic 30*, 315–342.

[2] For example, if ***R*** is to represent the space of all possible representations in general, then there will be no assumptions on the structure of **S**. In contrast, if ***R*** is to represent scientific reasoning, then one might presupposes that the elements in ***S*** obey fundamental physical laws.

[3] A reviewer suggested that weaker notions (e.g., a forgetful functor from category theory) might be more appropriate than permutation invariance for thinking about abstraction and formality. Future research should be done on a comparative or relative conception of permutation invariance such that one system of concepts can be more permutation invariant than another.

Syntactic Differentiation in Oscar Wilde's "Dorian Gray"

Melissa Wright, Reva Freedman
Northern Illinois University
DeKalb, Illinois 60115
meliswright16@gmail.com, rfreedman@niu.edu

Abstract

This study analyzes the syntax and constituents in Oscar Wilde's *The Picture of Dorian Gray*, looking for syntactic differences between quoted and narrative sections. We investigate sentence length, parse tree height and the frequency of various coordinating conjunctions in the text. We show that Wilde's character dialogue uses shorter sentences and less subordination than narrative passages. We also show frequency differences among the conjunctions studied. We hypothesize that these differences relate to the difference in working memory load between speaking and reading.

1 Introduction[1]

The aim of this study was to analyze differences in the syntactic form of sentences in two different types of running text. As a case study, we analyzed the syntactic structure of each sentence in Oscar Wilde's *The Picture of Dorian Gray* (1890). We distinguished between two types of text in the novel, dialogue and narrative. We analyzed three hypotheses: 1) that there was a difference in sentence length in the dialogue and narrative sections of the novel, 2) that there was a difference in the average height of parse trees, and 3) that there was a difference in the frequency of use of the conjunctions *and*, *but* and *or*.

A series of Python programs were created to clean up the input and make the text readable for the Stanford Parser[2] (Klein and Manning, 2003). The syntax trees provided by the parser enabled us to see how often certain parts of speech occurred and at what level of syntactic structure they were found. A CSV file was created with this information and fed to a final program to evaluate the hypotheses and calculate statistical significance.

2 Data Source and Cleanup

The input for this study, containing 57,673 words, was retrieved from the Project Gutenberg website.[3]

A few items had to be manually changed before preprocessing. The text contained poetry in both French and English, as well as entire paragraphs in French. Not only were these irrelevant to the hypotheses, but a parser trained on English prose could obviously not parse them. Furthermore, when these items were preceded by a colon (e.g., "He said: ..."), the colon had to be replaced by a period so that the introductory phrase would not appear to refer to the text after the deleted material, which might not even be a quote.

Since the Stanford Parser treats every period as final sentence punctuation, ellipses ("...") gave rise to empty sentences. As the material after an ellipsis can start with a capital letter (e.g., a proper noun), it was necessary to decide manually whether each ellipsis was intra-sentential, in which case it could be deleted, or terminal punctuation.

Following the manual phase, the second phase consisted of mechanized preprocessing. Periods and other terminators (e.g. !, ?, etc.) that occurred elsewhere than at the end of a sentence were dropped so that abbreviations such as "Mr." would not cause false sentence breaks. Similarly, we removed the periods that Oscar Wilde used after Roman numerals. Wilde's use of the unusual punctuation string ",--" was replaced by equivalent modern punctuation. Front and back matter, chapter headings (e.g., "CHAPTER XII"), and page numbers, formatted as [12], or [...12] when a chapter started mid-page, were also removed.

3 Methodology

The preprocessing phase enabled us to split sentences at the final punctuation mark. The text contained standalone quotations which were complete sentences, as well as narrative sentences which contained no quotations. However, it also contained sentences with an initial narrative portion introducing a quotation ("He said, '...'"), with a terminal narrative portion ("'...,' said Lord Henry"), or with a variety of more complex sentence structures, such as the following:

> "Oh, there is really very little to tell, Harry," answered the young painter; "and I am afraid you will hardly understand it. Perhaps you will hardly believe it."

To separate quoted from narrative material, we broke each sentence into segments, separating quoted and non-quoted material. Since the Stanford Parser parses sentences, a new

[2] http://nlp.stanford.edu/software/lex-parser.shtml.
[3] https://www.gutenberg.org/.

record was created when either a sentence break or a change in text type (from quote to non-dialogue or vice versa) occurred. For example, the above sentence was split into the following four segments:

Q: Oh, there is really very little to tell, Harry,
N: answered the young painter;
Q: and I am afraid you will hardly understand it.
Q: Perhaps you will hardly believe it.

We replaced segment-final punctuation with periods so that the parser would treat each segment as a complete sentence. As can be seen in the third segment above, or the segment "said Lord Henry", not all segments were sentences according to prescriptive English grammar; however, since the Stanford Parser is probabilistic, it could generally handle the segments we gave it. We verified some unusual cases before settling on this approach. For example, the parser could handle "said Lord Henry" but not "Said Lord Henry", which fortunately did not occur in the corpus.

Note that this decision does not affect the length or tree height of sentences in the quoted material but reduces both variables for narrative segments, since, for example, "answered the young painter" has a sentence length of 4 and contains no nested clauses, while the original sentence is obviously longer and contains coordination.

Figure 1 provides a sample output from the parser, showing examples of the extensive Penn Treebank tagset (Santorini, 1995). The parser output was used to calculate segment length, parse tree height and the frequency of *and*, *or* and *but*. We then conducted a two-tailed t-test with unequal variances on each of these variables, applying the Bonferroni correction in each case.

4 Results and Discussion

With regard to sentence length, the t-test showed that there is a significant difference ($p < .001$) in sentence length between the two types of text; narrative sentences are significantly longer. This result is even more striking when one considers that the length of the narrative segments was artificially depressed by the splitting mechanism employed.

Two extreme examples of sentence length are the two longest sentences in the book, both of which are pure narrative. One contains 198 words and the other contains 448 words. Although the rest of the book could be parsed in one batch, each of these sentences needed five times the default memory size of the Stanford Parser and had to be parsed separately.

```
(ROOT
  (S
    (NP (NNP Lord) (NNP Henry))
    (VP
      (VP (VBD stretched)
        (NP (PRP$ his) (JJ long) (NNS legs))
        (PRT (RP out))
        (PP (IN on)
          (NP (DT the) (NN divan))))
      (CC and)
      (VP (VBD shook)
        (PP (IN with)
          (NP (NN laughter)))))
    (. .)))
```

Figure 1: Sample parse from "The Picture of Dorian Gray"

Similarly, with regard to height of the syntax trees, the t-test showed that there is a significant difference ($p < .001$) in tree height between the two types of text; the narrative text also had significantly deeper trees. This finding is consistent with the previous one, providing further evidence that the narrative portion of the text is significantly more complex than the quoted portion.

This pattern could be due to Wilde's attempt to mirror real-world conversational style: if there are "limitations in human working memory and processing capacity [which] force reliance on a number of syntactic heuristics in order to make a provisional parse of a sentence as it is being processed" (Elman, 2009, p. 556), then it would be intuitive as an interlocutor to utilize less complex sentences within a discourse.

The findings illustrated in this study illustrate the importance of context when studying linguistic features. Within a conversation, there may be a subconscious expectation that speakers will utilize simpler constructions due to working memory load; however, when reading a descriptive passage in a written work, such limitations may not apply.

We examined three coordinating conjunctions, *and*, *but* and *or*. In the quoted text, *and* occurs approximately five times as often as *but*, while in the narrative text, it occurs approximately 25 times as often. Similarly, in the quoted text, *or* occurs approximately twice as often as *and*, while in the narrative text, these conjunctions have similar rates of occurrence. These data provide evidence that coordinating conjunctions play different roles in different types of text.

The relative rarity of *but* in the narrative text may result from the fact that *but* is most frequently used to show a contrast between two propositions. Therefore it would be less useful in longer sentences containing multiple propositions. *But* may also be more useful in dialogue, where it frequently occurs sentence-initially, allowing one character to signify disagreement with another. Further evidence from a text where *but* would be likely to occur in a non-quoted context, such as a deductive or argumentative context, would be useful in extending our understanding of *but*.

These patterns in Wilde's work may also display the author's linguistic thumbprint (Nolan, 2001). Previous research has been done on examining whether speakers and authors can be identified by unique linguistic patterns, and this study may indicate that there are indeed such patterns. Further evidence, from other works by Oscar Wilde and from other authors, would be required to evaluate this hypothesis.

5 Related Work

Coh-Metrix (Graesser et al., 2004) uses parts of speech, word frequency statistics and other features to measure cohesion. Wang et al. (2014) used syntactic features to differentiate conference papers from workshop papers. Freedman and Krieghbaum (2014) used syntactic features to differentiate early essays from revised ones by the same student authors, while Freedman and Krieghbaum (2015) used syntactic features to differentiate two faculty authors. Freedman (2017) used both syntactic and semantic means to differentiate sections of the book of Isaiah. Further information about the current study can be found in Wright (2017).

6 Conclusions

In this paper we analyzed the differences between quoted and narrative portions of Oscar Wilde's *The Picture of Dorian Gray*. Through parsing every sentence and analyzing the results, we showed that there are significant differences ($p < .001$) in both sentence length and parse tree height between quoted text and narrative portions. We also showed major differences in the frequency of various coordinating conjunctions between these two types of text.

Perhaps the most significant implication of this study is the possible intuition speakers and authors may have when taking part in or creating a conversation – if interlocutors instinctively know that shorter structures are to be employed in a conversational context, this may indicate underlying unconscious conversational syntactic principles.

The methods used in this study may enable future researchers to investigate linguistic components specific to an individual's written and oral speech patterns that will allow differentiation between personal style and unconscious conversational syntactic principles. Further research is also needed on the relationship between choice of conjunction, depth of clause embedding and discourse context, such as speaker quotation or other prose forms, including narrative, descriptive, explanatory and argumentative text.

There is also room for further analysis of the variation in sentence length between dialogue and narrative text. Such research has the potential to illustrate the strength of working memory load theories concerning real-time sentence processing, giving discourse psychologists a more solid foundation for investigating cognitive processing during conversation and reading.

Acknowledgments

Dr. Gulsat Aygen and Dr. Betty Birner, Department of English, Northern Illinois University provided advice on this study. The tree analysis code was based on code written by Doug Krieghbaum.

References

Elman, J. (2009). On the meaning of words and dinosaur bones: Lexical knowledge without a lexicon. Cognitive Science, 33: 547-582.

Freedman, R. 2017 (to appear). Can Natural Language Processing Help Identify the Author(s) of the Book of Isaiah? In Proceedings of the 30th Annual FLAIRS Conference.

Freedman, R., and Krieghbaum, D. 2014. Effects of Rewriting Essays on Linguistic Measures of Complexity. In 25th Annual Meeting of the Society for Text and Discourse.

Freedman, R., and Krieghbaum, D. 2015. Comparison of Expert Tutors through Syntactic Analysis of Transcripts. In Proceedings of 17th International Conference on Artificial Intelligence in Education (AIED 2015).

Graesser, A., McNamara, D., Louwerse, M., and Cai, Z. 2004. Coh-Metrix: Analysis of Text on Cohesion and Language. Behavior Research Methods, Instruments and Computers, 36: 193-202.

Klein, D., and Manning, C. 2003. Accurate Unlexicalized Parsing. In Proceedings of the 41st Meeting of the Association for Computational Linguistics, 423-430.

Nolan, F. 2001. Speaker identification evidence: Its forms, limitations, and roles. Proceedings of the conference "Law and Language: Prospect and Retrospect", December 12-15, 2001, Levi (Finnish Lapland). Available from http://www.ling.cam.ac.uk/francis/LawLang.doc.

Santorini, B. 1995. Part-of-Speech Tagging Guidelines for the Penn Treebank Project. University of Pennsylvania, Natural Language Processing Project. Technical Report MS-CIS-90-47, LINC LAB 178.

Wang, H., Di Eugenio, B., Lin, S., and Yu, C. 2014. Distinguishing Different Types of Conference Submissions: The ACL Case Study. Chicago Colloquium on Digital Humanities and Computer Science. Available at http://www.nlp.cs.uic.edu/PS-papers/dhcs_2014_manuscript.pdf.

Wilde, O. 1890. *The Picture of Dorian Gray*. From http://www.gutenberg.org/cache/epub/4078/pg4078.txt.

Wright, Melissa. 2017. Natural Language Processing and Syntactic Differentiation: A Corpus Case Study. M.A. thesis, Northern Illinois University.