Trust assessment in large-scale collaborative systems

Quang-Vinh Dang

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Évaluation de la confiance dans la collaboration à large échelle

∴ ∵ ∴

Trust assessment in large-scale collaborative systems

THÈSE

présentée et soutenue publiquement le 22 janvier 2018

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Quang-Vinh DANG

Composition du jury

Rapporteurs :	Sihem Amer-Yahia	Directrice de recherche, LIG/CNRS
	Wolfgang Prinz	Professeur, RWTH Aachen University
Examinateurs :	Isabelle Chrisment	Professeure, Université de Lorraine
	Lionel Brunie	Professeur, INSA de Lyon
Directeurs de thèse :	Francois Çharoy	Professeur, Université de Lorraine
	Claudia-Lavinia Ignat	Chargée de recherche, Inria Nancy Grand-Est

Institut National de Recherche en Informatique et en Automatique
Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Inria INVENTEURS DU MONDE NUMÉRIQUE

Loria Laboratoire lorrain de recherche en informatique et ses applications

Mis en page avec la classe thesul.

Acknowledgements

This thesis definitely is not possible without the great support from my supervisors, Dr. Claudia-Lavinia Ignat, Research Scientist at Inria Nancy Grand-Est and Dr. Francois Çharoy, Professor at Université de Lorraine. Deep from my heart, I would like to say thank you for your advice to keep me in a correct direction and your confidence to let me be free to discover new domains.

I would like to thank my PhD committee, Isabelle Chrisment, Sihem Amer-Yahia, Wolfgang Prinz and Lionel Brunie for accepting to be a member of the jury. It is my pleasure to present my thesis to them.

Personally, I would like to thank all members of COAST team, who make my time in Nancy so joyful and memorable.

Last but not least, thank my family who always support me unconditionally.

For Mom and Dad.

Résumé

Les systèmes collaboratifs à large échelle, où un grand nombre d'utilisateurs collaborent pour réaliser une tâche partagée, attirent beaucoup l'attention des milieux industriels et académiques. Bien que la confiance soit un facteur primordial pour le succès d'une telle collaboration, il est difficile pour les utilisateurs finaux d'évaluer manuellement le niveau de confiance envers chaque partenaire. Dans cette thèse, nous étudions le problème de l'évaluation de la confiance et cherchons à concevoir un modèle de confiance informatique dédiés aux systèmes collaboratifs.

Nos travaux s'organisent autour des trois questions de recherche suivantes.

1. **Quel est l'effet du déploiement d'un modèle de confiance et de la représentation aux utilisateurs des scores obtenus pour chaque partenaire ?** Nous avons conçu et organisé une expérience utilisateur basée sur le jeu de confiance qui est un protocole d'échange d'argent en environnement contrôlé dans lequel nous avons introduit des notes de confiance pour les utilisateurs. L'analyse détaillée du comportement des utilisateurs montre que: (i) la présentation d'un score de confiance aux utilisateurs encourage la collaboration entre eux de manière significative, et ce, à un niveau similaire à celui de l'affichage du surnom des participants, et (ii) les utilisateurs se conforment au score de confiance dans leur prise de décision concernant l'échange monétaire. Les résultats suggèrent donc qu'un modèle de confiance peut être déployé dans les systèmes collaboratifs afin d'assister les utilisateurs.

2. **Comment calculer le score de confiance entre des utilisateurs qui ont déjà collaboré ?** Nous avons conçu un modèle de confiance pour les jeux de confiance répétés qui calcule les scores de confiance des utilisateurs en fonction de leur comportement passé. Nous avons validé notre modèle de confiance en relativement à: (i) des données simulées, (ii) de l'opinion humaine et (iii) des données expérimentales réelles. Nous avons appliqué notre modèle de confiance à Wikipédia en utilisant la qualité des articles de Wikipédia comme mesure de contribution. Nous avons proposé trois algorithmes d'apprentissage automatique pour évaluer la qualité des articles de Wikipédia: l'un est basé sur une forêt d'arbres décisionnels tandis que les deux autres sont basés sur des méthodes d'apprentissage profond.

3. **Comment prédire la relation de confiance entre des utilisateurs qui n'ont pas encore interagi ?** Etant donné un réseau dans lequel les liens représentent les relations de confiance/défiance entre utilisateurs, nous cherchons à prévoir les relations futures. Nous avons proposé un algorithme qui prend en compte les informations temporelles relatives à l'établissement des liens dans le réseau pour prédire la relation future de confiance/défiance des utilisateurs. L'algorithme proposé surpasse les approches de la littérature pour des jeux de données réels provenant de réseaux sociaux dirigés et signés.

Mots-clés: collaboration, confiance, théorie des jeux, apprentissage automatique

Abstract

Large-scale collaborative systems wherein a large number of users collaborate to perform a shared task attract a lot of attention from both academic and industry. Trust is an important factor for the success of a large-scale collaboration. It is difficult for end-users to manually assess the trust level of each partner in this collaboration. We study the trust assessment problem and aim to design a computational trust model for collaborative systems.

We focused on three research questions.

1. **What is the effect of deploying a trust model and showing trust scores of partners to users?** We designed and organized a user-experiment based on trust game, a well-known money-exchange lab-control protocol, wherein we introduced user trust scores. Our comprehensive analysis on user behavior proved that: (i) showing trust score to users encourages collaboration between them significantly at a similar level with showing nickname, and (ii) users follow the trust score in decision-making. The results suggest that a trust model can be deployed in collaborative systems to assist users.

2. **How to calculate trust score between users that experienced a collaboration?** We designed a trust model for repeated trust game that computes user trust scores based on their past behavior. We validated our trust model against: (i) simulated data, (ii) human opinion, and (iii) real-world experimental data. We extended our trust model to Wikipedia based on user contributions to the quality of the edited Wikipedia articles. We proposed three machine learning approaches to assess the quality of Wikipedia articles: the first one based on random forest with manually-designed features while the other two ones based on deep learning methods.

3. **How to predict trust relation between users that did not interact in the past?** Given a network in which the links represent the trust/distrust relations between users, we aim to predict future relations. We proposed an algorithm that takes into account the established time information of the links in the network to predict future user trust/distrust relationships. Our algorithm outperforms state-of-the-art approaches on real-world signed directed social network datasets.

Keywords: collaboration, trust, game theory, machine learning

Contents

List of Figures

Chapter 1

Introduction

Man is by nature a social animal

— Aristotle, Politics

Contents

1.1 Research Context

Collaboration is defined in Oxford Advanced Learner's Dictionary as "the act of working with another person or group of people to create or produce something" [Sally et al., 2015].

Human societies might not have been formed without collaboration between individuals. Human need to collaborate when they can not finish a task alone [Tomasello et al., 2012]. Kim

Hill, a social anthropologist at Arizona State University, stated that "humans are not special because of their big brains. That's not the reason we can build rocket ships – no individual can. We have rockets because 10,000 individuals cooperate in producing the information" [Wade, 2011]. Collaboration is an essential factor for the success in the 21^{st} century [Morel, 2014].

Before the Internet era, collaboration was usually formed within small groups whose members were physically co-located and knew each other. Studies [Erickson and Gratton, 2007] argued that in 20^{th} century "true teams rarely had more than 20 members" . According to the same research study, today "many complex tasks involve teams of 100 or more". Collaboration from distance is easier for everyone thanks to the Internet.

Collaborative systems are the software systems which allow multiple users to collaborate. Some collaborative systems today are *collaborative editing systems*. They allow multiple users who are not co-located to share and edit documents over the Internet [Lv et al., 2016]. The term "document" can refer to different kinds of document such as a plain text document [Gobby, 2017], a rich-text document like in Google Docs [Attebury et al., 2013], a UML diagram [Sparx, 2017] or a picture [J. C. Tang and Minneman, 1991]. Other examples of collaborative systems are collaborative e-learning systems where students and teachers collaborate for knowledge sharing [Monahan et al., 2008].

The importance of collaborative systems is increasing over recent years. An evidence is that the collaborative systems attract a lot of attention from both academy and industry, and their number of users has increased significantly over time. For example, we display the number of users of ShareLatex, a collaborative Latex editing system, over last five years in Figure 1.1. The number of users of ShareLatex increases rapidly. Zoho[1] - a collaborative editing system similar to Google Docs - achieved the number of registered users of 13 millions [Vaca, 2015]. The number of authors who collaborated in scientific writing has increased over years as displayed in Figure 1.2. Collaboration is more and more popular in scientific writing [Jang et al., 2016; Science et al., 2017]. Version control systems like git and their hosting services such as Github became de-facto standard for developers to share and collaborate [Gerber and Craig, 2015]. In April 2017, Github has 20 millions registered users and 57 millions repositories [Firestine, 2017].

In traditional software systems such as Microsoft Office[2], users use and interact with the software system only. In collaborative systems, user need to interact not only with the system but also with other users. Therefore, the usage of collaborative systems raises several new issues that will be discussed in the next section.

In the following section we discuss about the new issues of collaborative systems. Then we discuss about trust between human in collaboration as our research topic. Afterwards we formalize our research questions, present related studies and our contributions for each research question.

1.1.1 Issues of collaborative systems

In a collaborative system, a user needs to use the system and interact with other users called *partners* in this thesis.

Studies [Greenhalgh, 1997] indicated several problems in developing collaborative systems. These problems are similar with problems in developing traditional software systems, such as designing a user interface for collaborative systems [J. C. Tang and Minneman, 1991; Dewan and Choudhary, 1991], improve response time [R. Kraut et al., 1992] or designing effective merging algorithms that combine modification of users [C.-L. Ignat et al., 2017]. Collaborative systems

[1] https://www.zoho.com/

[2] We refer to the desktop version, not Office 365 where users can collaborate online.

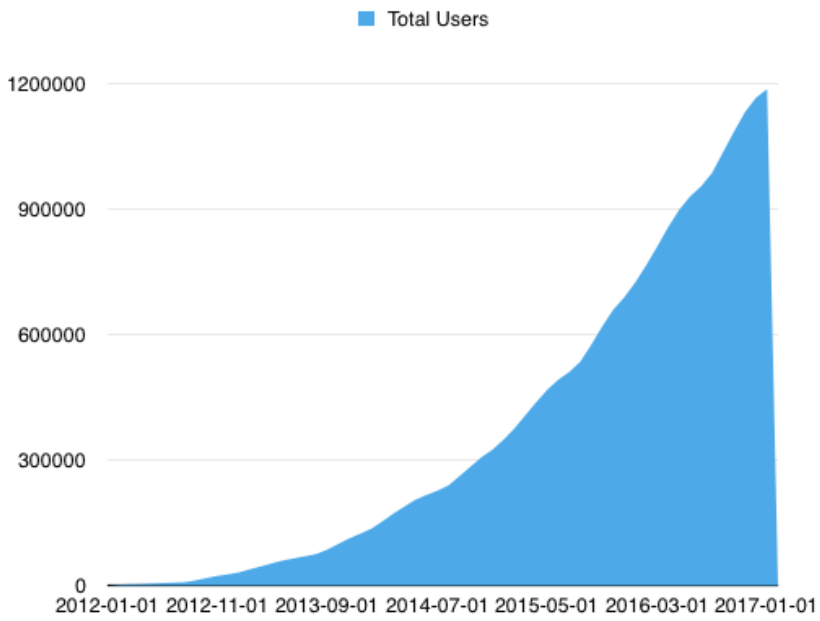


Figure 1.1: Number of ShareLatex's users over years. Image source: [ShareLatex, 2017].

like Google Docs are widely used in small-scale [Tan and Y. Kim, 2015]. Surveys and user experiments [Edwards, 2011; Wood, 2011] claimed the positive perception from Google Docs users.

However, in collaborative systems, users interact with their partners to finish tasks. We assume that the main objective of a user is to finish tasks at the highest quality level. The final outcome depends not only on the user herself but also all her partners. If a *malicious* partner is accepted to join a group of users and is able to modify the shared resource, she can harm other *honest* users. We define malicious users as users performed malicious actions.

The malicious actions can take different forms in different collaborative systems. In Wikipedia, malicious users can try to insert false information to attack other people or promote themselves. These modifications are called *vandalism* in Wikipedia [Potthast et al., 2008; P. S. Adler and C. X. Chen, 2011]. In source-code version control system such as git, malicious users can destroy legacy code or insert virus into the code [B. Chen and Curtmola, 2014]. Git supports revert action but it is not easy by non-experienced users [Chacon and Straub, 2014]. In collaborative editing systems such as ShareLatex, a malicious user can take the content written by honest users for an improper usage, such as to use the content in a different article and claim their authorship.

Alternatively, if a user collaborates with honest partners, they can achieve some outcomes that no individual effort can. The claim has been confirmed by studies in different fields [Persson et al., 2004; Choi et al., 2016], such as in programming [Nosek, 1998] or in scientific research [Sonnenwald, 2007]. For instance, it is popular in scientific writing today that a scientific article is written by multiple authors [Science et al., 2017; Jang et al., 2016] because each author holds a part of the knowledge which is needed for the article. If they can collaborate effectively together they can produce a scientific publication. Otherwise each of them only keeps a meaningless piece of information. In collaborative software development, it is often that

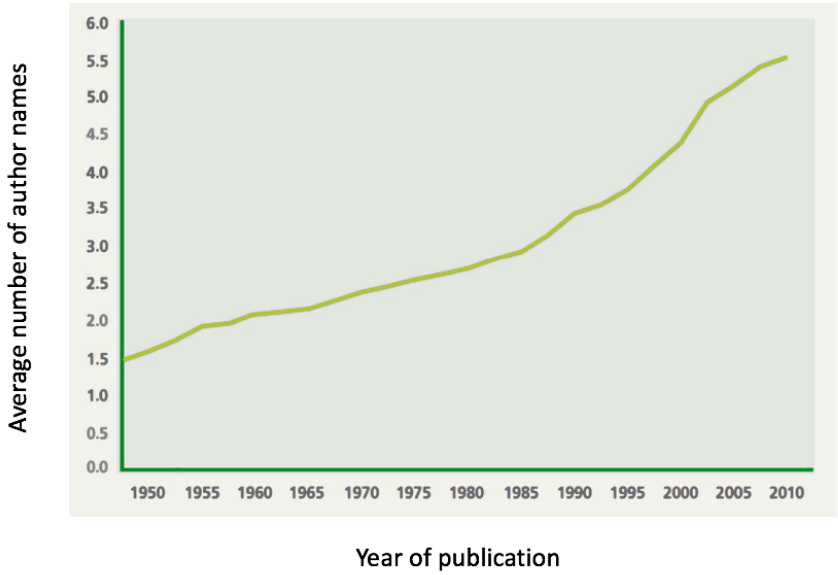


Figure 1.2: Average number of collective author names per MEDLINE/PubMed citation (when collective author names present). Image source: [Science et al., 2017].

developers in the team have expertise in a narrow field. For instance a developer has experience in back-end programming while another developer only has knowledge in user interface design and implementation. If these two developers do not collaborate with each other, none of them can build a complete software system.

In collaborative systems, a user decides to collaborate with a partner or not by granting some rights to the partner. For instance, in Google Docs or ShareLatex, the user decides to allow a partner to view and modify a particular document or not. In git repositories, the user decides to allow a partner to view and modify code. The user needs to make a right decision, i.e. to collaborate with honest partners and not with malicious ones.

However, we only can determine malicious partners if:

- Malicious actions have been performed.
- The user is aware about the malicious actions. For instance, the user needs to be aware about the actions, or the direct or indirect consequences of the actions. If the user is aware of a potential malicious action, she also needs to decide if this action is really a malicious action or just a mistake [Avizienis et al., 2004]. Therefore, usually a single harmful action is not enough to determine one partner as a malicious partner.

As an example, suppose Alice collaborates with Bob and Carol. Bob is a honest partner and Carol is a malicious one. However, so far both Bob and Carol collaborated and none of them performed any malicious activity. The malicious action is only planned inside Carol's mind. In this case, there is no way for Alice to detect Carol as a malicious user unless Alice can read Carol's mind which is not yet possible at the time of writing [Poldrack, 2017]. Furthermore, if Carol performed the malicious action but the result of this action has not been revealed to Alice, Alice also cannot detect the malicious partner.

Unfortunately, it is usual in collaborative systems that the user can reveal the result of a malicious action after a long time. In some cases, the results will never be revealed.

Suppose Alice is a director of an university and she inserted a wrong information into Wikipedia to claim that her university is the best one in the continent with modern facilities and a lot of successful students. The result might be that the university attracts more student, receives more supporting fund or be able to recruit better researchers - but these results might take a long duration or even are impossible to reveal. As of this writing, it is not easy to detect wrong information automatically [Y. Zheng et al., 2017]. Some Wikipedia editors received money to insert wrong or controversial information [Pinsker, 2015].

The bad outcomes might also come from the fact that partners lack competency, i.e. they do not have enough information or skill to finish the task with an expected quality. For instance, a developer might insert an exploiting code without intention. It might be difficult to distinguish whether the action was malicious. However as we discuss in Section 1.1.2, a user might not need to distinguish a malicious action from an unintended one. The reason is that trust reflects the user expectation that a partner adopts a particular kind of behavior in the future.

Hence the user has to decide to collaborate with a partner or not with some uncertainty about future behavior of this partner. Moreover the results of future behavior are also uncertain. In other words, there is *risk* in collaboration. To start the collaboration, the user needs to *trust* their partner at a certain level.

1.1.2 Trust as Research Topic

Studies claimed that trust between humans is an essential factor for a successful collaboration [Mertz, 2013]. [Cohen and Mankin, 1999, page 1] defined virtual teams as team "composed of geographically dispersed organizational members". We can use the definition to refer to the team who collaborate using a collaborative system over the Internet and some members of the team do not know each other. [Kasper-Fuehrera and Ashkanasy, 2001; L. M. Peters and Manz, 2007] claimed that trust is a vital factor for the effectiveness of the virtual teams.

Because trust is a common and important concept in different domains, the term has been defined in different ways and there is no wide-accepted definition [Rousseau et al., 1998; Cho et al., 2015].

In psychology, trust is defined as "an expectancy held by an individual that the word, promise, verbal or written statement of another individual can be relied upon" [Rotter, 1967, page 651] or "cognitive learning process obtained from social experiences based on the consequences of trusting behaviors" [Cho et al., 2015, page 3]. [Rousseau et al., 1998, page 395] reviewed different studies on trust and proposed a definition of trust as "a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behavior of another". The definitions of [Rotter, 1967] and [Rousseau et al., 1998] focus on the *future expectation* of trust, while the definition presented in [Cho et al., 2015] focused on the *historical experience* of trust: trust is built based on observations in the past.

In sociology, trust is defined as "subjective probability that another party will perform an action that will not hurt my interest under uncertainty and ignorance" by [Gambetta, 1988, page 217] while [Sztompka, 1999, page 25] defined trust as "a bet about the future contingent actions of a trustee". The trust definition in sociology emphasizes the *uncertainty* aspect of trust: people need to trust because they do not know everything.

In computer science, the definition of trust is derived from psychology and sociology [Sherchan et al., 2013] and is given as "a subjective expectation an entity has about another's future behavior" [Mui, 2002, page 75].

The definitions of trust in literature are diverse. However they share some similarities. Based on the above definitions, we can address some features of trust relations. When a user trusts a

partner, it means:

- The user expects that the partner will behave well in the future. As we discussed in the previous section, the definitions of *well behavior* are different in different settings, depending user objectives. For example, in Wikipedia a user could expect that a partner will not insert a wrong information, while in github a user could expect that a partner will not insert a virus code to the code repository.
- The user accepts the risk that a partner might perform a malicious activity. It means, trust is only needed in the presence of risk [Mayer et al., 1995].
- The trust assessment is based on historical experience of the user with the partner [Denning, 1993].
  - Based on this feature we can state that trust depends on the context, i.e. a user could trust a partner in doing a particular task but not in doing another task, because the user only observed the behavior of the partner in the first task but not in the second one. For instance, Alice trusts Bob in writing code because she observed Bob doing implementation in the past, but it does not mean that Alice trusts Bob in drawing UML diagrams.
  - As we briefly mentioned in the previous section, a partner can perform a harmful activity with or without intention. The user can not know the intention of the partner. The user only can observe the behavior of the partner to decide the trustworthiness of this partner.

From the above definitions of trust, we claim that trust is a *personal* state [Cho et al., 2015] because trust is based on personal experience of a user on a partner. Therefore, we distinguish *trust* and *reputation*. Trust reflects personal opinions, i.e. Alice trusts Bob, while reputation reflects collective opinions from a community to a person [Ruan and Durresi, 2016]. Usually higher reputation leads to higher trust [Doney and Cannon, 1997] but this claim is not necessary true: even Bob is well-considered by the community, Alice personally might not trust him because her experience with Bob is different from other people. In other words, trust is an *one-to-one* relation [Abdul-Rahman and Hailes, 1997] while reputation is a *many-to-one* relation.

Trust is one of the most critical issues in general online systems where users do not have much information about each other [Golbeck, 2009]. If users have no trust in their partners collaboration becomes very difficult. In many cases, there will be no activity to be performed if the trust level between users is too low [Dasgupta, 2000]. As an example, in e-commerce systems the lack of trust is one of the most popular reasons for consumers not buying [M. K. O. Lee and Turban, 2001]. Before collaborating with a partner, a user should be able to assess the trust level of their partners.

Suppose Alice is writing a scientific article on ShareLatex and Bob asks to join the project. Alice needs to decide to accept the request of Bob or not. In order to do that, she assess the trust level of Bob to evaluate the expectation and the risk. Alice could perform the trust assessment by two main approaches [Cho et al., 2015]: she can assess the trust level of Bob by reviewing her own experience with Bob, or she can do that by evaluating the indirect relations between her and Bob, e.g. if she does not know Bob well but she trusts Carol and Carol trusts Bob, Alice could trust Bob also [Guha et al., 2004]. If the risk is too high, Alice will not collaborate with Bob.

As previously mentioned, besides technical hardware/software issues, usage of collaborative systems is challenging. There is not yet comprehensive studies about the user-related problems and particularly the problem of trust assessment between users in collaborative contexts.

In common sense, trust is a fuzzy concept [McKnight and Chervany, 2001]. One could believe that trust is neither measurable nor comparable [S. P. Marsh, 1994]. For instance, in daily life it is rare to hear Alice stating that she trusts Bob at 62.4%. However, various studies [Thurstone, 1928; Mui et al., 2002; Golbeck, 2009; Brülhart and Usunier, 2012; Sherchan et al., 2013; Hoelz and Ralha, 2015] claimed that trust can be measured, i.e. trust level between users can be represented by numerical values. A computational trust model can be designed to calculate trust level between users.

In the next section, we discuss the need of computational trust models in large-scale collaborative systems.

1.2 Research Questions

As we discussed in the previous section, trust assessment is important in collaborative systems. However, people are using collaborative systems such as Google Docs without a trust assessment tool. Thereupon someone could ask why should we introduce the idea of trust models and trust scores to users.

Most collaborative systems only support small-scale collaboration, i.e. they allow a small number of users to share a document. For instance, Google Docs [Google, 2017] or Dropbox Paper [Center, 2017] allows up to 50 users to edit a document at the same time. In practice Google service might stop when the number of users reaches to 30 [Q. Dang and C. Ignat, 2016c]. In scientific writing, the average number of authors of a scientific article is around 5 [Science et al., 2017; Economist, 2016; Jang et al., 2016]. Nevertheless, studies addressed the need of large-scale collaboration where the number of users can reach thousands or more [Richardson and Domingos, 2003; Elliott, 2007]. For instance, the average number of authors for an article is increasing over years [Jang et al., 2016]. There are scientific articles which are the result of a collaborative work between five thousands scientists [Castelvecchi, 2015]. Wikipedia and Linux kernel project are well-known examples of large-scale collaboration where the number of users reaches to millions [Doan et al., 2010].

We distinguish *large-scale* collaborative systems with *small-scale* systems by the number of users. However, to the best of our knowledge, there is not yet a clear distinction between *large-scale* collaboration and *small-scale* collaboration in literature, despite the fact that the term "large-scale collaboration" has been mentioned several times in research studies [Gaver and R. B. Smith, 1990; Gu et al., 2007; Siangliulue et al., 2016].

Researchers used the term *large-scale collaboration* to refer to various collaboration sizes. [Star and Ruhleder, 1994] studied the collaboration of $1,400$ geneticists from over 100 laboratories. [P. S. Adler and C. X. Chen, 2011] considered an example of a collaboration between 5000 engineers in designing a new aircraft engine as a large-scale collaboration. [Kolowich, 2013] reported a case when the number of users in real-time collaborative editing systems reaches tens of thousands, which definitely overcame the supported limit size causing system break. We can consider the collaboration on Github as large-scale collaboration. Studies [Thung et al., 2013] stated that it is common for a Github developer contributes to a same project together with more than $1,000$ other developers.

In small-scale collaborations, users can assess the trust level of their partners by remembering and recalling their experience with these partners [Teacy et al., 2006]. In large-scale collabora-

tions where the number of users is huge, it is difficult for a user to recall and analyze their history in order to assess the trust level of a particular partner among other partners. [Abdul-Rahman and Hailes, 1997] claimed that it is not possible for an average user to analyze the potential risk of every on-line interaction. Furthermore, [Riegelsberger, Martina Angela Sasse, et al., 2005, page 405] specifically notes the overhead associated with the maintenance of partner-specific trust values. Therefore, users need assistance in assessing the trustworthiness of their partners.

Different techniques have been used to allow users to judge the trustworthiness of their partners [Grabner-Kraeuter, 2002; Clemons et al., 2016]. Websites today rely on several mechanisms which are reputation score [Gary E Bolton et al., 2002], nick-name or ID[3] [Corbitt et al., 2003; Jøsang, Fabre, et al., 2005], avatar [Yuksel et al., 2017] and review [Park et al., 2007] to support users in deciding to trust another user or not.

Each of the above methods have their shortcomings. We will discuss them in details in Section 2.1. Reputation schemes and review systems are vulnerable to attacks from malicious third parties [Hoffman et al., 2009], while identity and avatar can be faked or changed easily. Furthermore, review, identity and avatar do not scale well.

Studies [Abdul-Rahman and Hailes, 1997; Golbeck, 2009] suggested that a computational trust model can be deployed to assist users in assessing the trustworthiness of their partners so they can decide to collaborate with this partner or not.

The task of a trust model is to calculate and display the computational trust level of a partner to a user. The value can be in a form of binary-trust level, i.e. trust/distrust relations [Golbeck and Hendler, 2006; Leskovec et al., 2010a] or in a form of a numerical value [Abdul-Rahman and Hailes, 1997; Xiong and L. Liu, 2004].

Using a computational trust model a user can calculate trust score of other partners by using only the information she observed. The user does not need to rely any external information. Hence it is more difficult to attack trust score compared to other techniques.

A trust model has several advantages compared to other mechanisms:

- It is easy to use. Users do not need to remember anything as opposed to identity or avatar.
- It does not require a central server. Any user can compute a trust score by herself without querying an external information.
- It cannot be modified by third-party. Therefore trust score is robust against many attacks which are available to reputation schemes. We will discuss more about this in Section 2.1.1.3.

To the best of our knowledge there is not yet a study that verified quantitatively the effect of a trust model to user behavior in collaboration. Moreover, the problem of designing computational trust models for collaborative systems has not been studied comprehensively.

In this thesis we study the computational trust models for large-scale collaborative systems. We will focus on three research questions:

1. Should we deploy a computational trust model and display trust score of partners to the users? In other words, does the fact that the trust scores of partners are displayed to users has effect on user behavior?
2. If a trust model is useful, how do we calculate trust score of users who collaborated?

[3] In this thesis we used the term *nick-name* and *ID* interchangeably, refer to a unique virtual identity associated with a user account on a website.

3. In case users did not interact with each other, can we predict future trust/distrust relations between them?

In the following we will discuss in details each research question.

1.2.1 Should we introduce trust score to users?

As of this writing we are not aware of any real-world systems that integrated a computational trust model. Therefore, we do not know the effect of deploying a trust model and display trust scores on user behavior.

As [Franklin Jr, 1997, page 74] stated, "even perfect technology solutions are useless if no one can be persuaded to try them". The need of computational trust models has been addressed for a long time [Abdul-Rahman and Hailes, 1997]. To the best of our knowledge, no study focusing on the influence of a computational trust model on user behavior.

Particularly in collaborative contexts, we do not know if introducing the trust score to users will encourage the collaboration between them. We do not know if the users will notice and follow the guidance of trust score, i.e. they will prefer to collaborate with high score partners or not. We will address these problems in the first part of this thesis.

1.2.2 How do we calculate the trust score of partners who collaborated?

The second research question is how to calculate trust score of partners?

Assume that in a particular collaborative system Alice considers to collaborate with Bob and she wants to calculate her trust score on Bob. Studies have proposed several ways to assess trust [Jiang et al., 2016]. Most of them rely on external information, i.e. if Alice wants to assess the trustworthiness of Bob, she has to query some information from other members say Carol or Dave [Jøsang, S. Marsh, et al., 2006; R. Zhang and Y. Mao, 2014]. These external information needs to be verified to make sure that Alice does not receive the wrong information [Jøsang, S. Marsh, et al., 2006]. Furthermore, this information is not always available, e.g. Dave might not want to tell Alice what he thinks about Bob.

In fact, the most reliable information Alice can rely on is the one observed by herself in the system. We call the information about historical observation of a user as *history log* in this thesis. For instance, in Google Docs, Alice can rely on the activity log of documents that she can access. The computational trust models should calculate the trust score of Alice on Bob using only this history log.

We defined the second research question as: in a particular context, assuming the history log of a user A is available, how do we calculate the trust score of A on a partner B.

Different collaboration contexts require different trust calculation methods [Huynh, 2009; Pinyol and Sabater-Mir, 2013]. The reason is that in different contexts, the definition of collaboration or malicious actions as well as gain or loss for users are different. Due to the fact that several collaboration systems are available today, it is not possible to cover all of them within the scope of this thesis. We will focus on two selected contexts which are Wikipedia and repeated trust game to study the computational trust models. These contexts will be discussed in Section 1.3.

1.2.3 How do we predict the trust/distrust relations of users who did not interact with each other?

We address the problem of calculating trust score for users who have already interacted in the second research question. In the third research question, we will focus on the relationship between users who did not interact with each other. In large-scale collaborative systems, the number of partners that a user collaborated with is usually a very small number compare to the total number of users in the system [Laniado and Tasso, 2011; Thung et al., 2013].

At some points of time a user will need to extend their network and setup collaboration with a partner that she did not interact with before. For instance, Alice is maintaining a project on Github. Bob discovered the project through the Internet and he wants to join the project. However, Alice does not know Bob, but she needs to decide to accept Bob joining the project or not. In this situation, because there is no interaction between two users, calculating trust score as in the previous section is not possible [X. Liu et al., 2013].

Studies [Guha et al., 2004; J. Tang, Y. Chang, et al., 2016] suggested that, if the information of trust/distrust relationship between a subset of users is provided, we can predict the trust/distrust relationship between two users who never interacted with each other before. Therefore, we can recommend a user to trust or not a particular partner. However, due to the lack of information, we can only provide binary-trust level recommendation, i.e. we can only predict the future trust/distrust relationship between two users.

We address the research question in this thesis: "How to predict a particular future relationship from a user to a partner as trust or distrust, given the relationship between other pair of users?" [Leskovec et al., 2010a].

1.3 Study Contexts

As we discussed in Section 1.1, many collaborative systems are available today. In this thesis, we will focus on two contexts which are Wikipedia and repeated trust game to address the research questions defined in Section 1.2. In what follows we review these two contexts.

1.3.1 Wikipedia

Wikipedia is "a free online encyclopedia that, by default, allows its users to edit any article" [Wales and Sanger, 2001]. Different from traditional encyclopedia such as Britannica whose authors are well-known scholars, the content of Wikipedia is created by a huge number of contributors, mostly unknown and volunteering, from all over the world. Wikipedia contributors (or Wikipedians) can also vote *for* or vote *against* other contributors to elect them to be administrators of particular Wikipedia pages in the process called *Request for Adminship* (RfA) [Burke and R. E. Kraut, 2008]. Wikipedia is built based on a collaboration system called Wiki [Wikipedia, 2017e]. Wikipedia is the largest and probably one of the most important Wiki-based systems in the world [Laniado and Tasso, 2011; Zha et al., 2016].

Wikipedia is the result of an incredible collaboration between millions of people. A Wikipedia editor [Nov, 2007] can positively contribute to Wikipedia by adding content, fixing errors or removing irrelevant text [J. Liu and Ram, 2011] but also can destroy the value of Wikipedia by removing good content or adding advertisements to promote herself. These actions are called vandalism [Potthast et al., 2008; Tramullas et al., 2016]. A Wikipedian can deviate to gain her own benefit: studies suggested that people have a lot of motivations to contribute and claim their ownership of Wikipedia content [Forte and Bruckman, 2005; Kuznetsov, 2006].

We chose Wikipedia because of two reasons.

The first reason is that Wikipedia probably is the result of the most important collaboration today. Wikipedia is the dominant information source for the entire generation of Internet users [Brown, 2011]. Modern users tend to take for granted information from Wikipedia even regarding health-care information [Jones, 2009]. This phenomenon is more popular among youths [Pan et al., 2007; Rowlands et al., 2008]. Furthermore, users of popular search engines such as Google usually reach Wikipedia [Natalie Kupferberg et al., 2011], increasing the influence of information presented on Wikipedia.

The second reason is that Wikipedia provides a well-annotated open datasets which allows us to evaluate the quality of our proposed trust model. As we will discuss in Section 1.4.2.2, our computational trust models are based on the *quality* of the previous contributions of partners. It is not trivial to determine the quality of contributions in Wikipedia, and we need to design some algorithms to predict their quality. Wikipedia provides a large set of articles with quality labels assigned manually by Wikipedia reviewers, and these quality labels are officially approved by Wikipedia [Warncke-Wang, Cosley, et al., 2013]. Therefore we can train and test our algorithms in predicting the quality of articles and then to measure the quality of each individual contribution.

Annotated datasets with quality level are not available in other popular collaboration systems. To the best of our knowledge, there is not yet a well-accepted definition of *contribution quality* in other collaborative systems. However, as we will discuss in next chapters, the ideas of our trust model for Wikipedia can be easily extended to other systems.

Furthermore, datasets where Wikipedia editors explicitly express their trust/distrust opinions on other editors in RfA process are available [Burke and R. E. Kraut, 2008; Leskovec et al., 2010b]. These datasets allow us to train and validate our trust/distrust prediction algorithm. There is no dataset with trust annotation for other systems such as Github [Cruz et al., 2016] making impossible to validate the algorithm.

As we will describe in following chapters, the trust models and trust/distrust relationship prediction algorithms are validated against not only Wikipedia dataset but also other datasets collected from different time and location. The results allow us to be confident that the ideas of our proposed algorithms can be applied not only in Wikipedia but in other collaborative contexts.

1.3.2 Collaborative Games

Collaborative games are games wherein multiple players need to collaborate to achieve the best collective payoff [Riegelsberger, M Angela Sasse, et al., 2003]. These games are usually game-theoretic protocols. They are widely used in psychology, experimental economic and behavioral studies to conduct research about human behavior [Chakravarty et al., 2011], but also can benefit research studies in computer science [Grossklags, 2007] or in computer-human interaction [Nguyen and Canny, 2007].

A very popular collaborative game is the prisoner-dilemma [Tucker, 1950]. In this game, if two players collaborate, they will achieve the highest collective payoff. However, each player always has incentive to deviate, and it is very difficult to form the collaboration. Game theory predicts that, in one-shot prisoner dilemma, two players will both deviate [Camerer, 2003, Chapter 2]. Prisoner dilemma has been used under various conditions [Murnighan and L. Wang, 2016]. Different techniques have been proposed to encourage users to collaborate, but the most common way is to allow repeated prisoner dilemma experiments [Kendall et al., 2007].

In traditional settings of prisoner dilemma experiment, two players make decision simultaneously. [Riegelsberger, M Angela Sasse, et al., 2003] suggested that the prisoner dilemma is too limited in studying human trust because of two reasons: (1) it covers a very specific subset of trust-requiring situations, and (2) it does not take all sources of vulnerability into account. The authors suggested to use sequential experiments instead of prisoner-dilemma.

Players in prisoner dilemma have only two choices that are collaboration or deviation. Studies suggested that trust value might be a continuous value rather than a simple binary decision [Xiong and L. Liu, 2004; S. Marsh and Briggs, 2009]. [Berg et al., 1995] presented an extension of sequential prisoner dilemma called *trust game*[4], wherein the players can select an arbitrary number of action within a range. In trust game, trust between players can be measured more precisely [Glaeser et al., 2000; Brülhart and Usunier, 2012].

Trust game is a game between two players: *sender* and *receiver*. An one-trial trust game contains two turns. First, the senders sends an amount between 0 and 10 to the receiver. Suppose the sending amount is x. The receiver will receive $3 * x$ on their side. In the second turn, the receiver sends an amount y between 0 and $3 * x$ back to the sender. In this turn, the sender receives y to their balance. The game can be repeated, i.e. the game can be played in multiple rounds and the roles of players can be changed [Cochard et al., 2004]. We will use *repeated trust game* in this thesis.

Similar to prisoner-dilemma, in trust game the highest payoff will be maximized if two players collaborate, i.e. if the sender sends 10 and the receiver sends back an amount which is large enough to maintain the future collaboration. However each player has the incentive to deviate for maximizing their own profit, i.e. a player can deviate by sending 0 to maximize their own profit in this round while reducing the profit of their partner [Camerer, 2003]. By doing so they also destroy the future collaboration .

Trust game is used as one of our study context due to several reasons:

- User experimental protocols like trust game are important research tools to understand human behavior [Brandenburger and Nalebuff, 2011]. They provide a general guideline to design real-world systems. In fact, experimental games, along with surveys, are two main measurement strategies in studying trust [Dinesen and Bekkers, 2015]. Human-Computer Interaction (HCI) studies have used games like prisoner-dilemma and trust game for a long time to study how do users trust each other under different conditions [Riegelsberger, M Angela Sasse, et al., 2003].

  Various studies [Falk and Heckman, 2009; Charness and Kuhn, 2011] suggested that the results from lab-control experiments can be applied successfully into real-world systems. For instance, [Yao and Darwen, 1999] suggested that the reputation score can encourage users to collaborate more while deviate less. This phenomenon has been confirmed on eBay [Resnick, Zeckhauser, et al., 2006]. The role of avatar in increasing trust has been confirmed in Second Life [Hemp, 2006] and in experiments [Bente, Dratsch, Rehbach, et al., 2014]. [Laaksonen et al., 2009] used the trust game to analyze the interfirm trust with data collected from interviews with managers. The main message is that, there is a consistency between findings in lab-control experiments and human behavior in real life.

[4] In fact, Berg called their game as *investment game*, but many follow-up studies used the term *trust game* [Johnson and Mislin, 2011; Murnighan and L. Wang, 2016; Cooper and Kagel, 2016], while several research works used the term *trust game* to refer the sequential prisoner dilemma setting [Riegelsberger, M Angela Sasse, et al., 2003; Rabanal and Friedman, 2015]. To be consistent, in this thesis, we used *trust game* to refer the game presented by [Berg et al., 1995], and *sequential prisoner dilemma* for the other game.

Therefore, we could expect that effects of trust score on user behavior in repeated trust game will be found in real-world collaborative settings.

In this thesis, we propose a computational trust model for repeated trust game. As we will discuss in Section 3.2.5, the trust model can be applied to calculate trust of Wikipedians. The requirement is that we need to propose a method to convert the behavior of Wikipedians into numerical values so we can apply the trust model of trust game.

- In trust game, researchers can easily control the condition of experiments, i.e. we can change only one setting while keeping other settings constant. It is not easy to do that in real-world systems.
- Studies showed that the exchanging amounts between users reflect their trust on each others [Brülhart and Usunier, 2012]. It is an important feature of trust game, because the representation of trust of users on partners might be not clear in other real-world settings. Hence, using trust game we can measure the trust between users by their behaviors [Glaeser et al., 2000] under different conditions.
- Values like gain and loss of users as well as options of users are well-designed because they are represented by numerical values already [Rapoport, 1973], making the results easy to analyze. For instance, it is not trivial to define the gain and loss of users who contribute to Wikipedia.

As we will discuss in following chapters, the trust model we presented for repeated trust game can be applied to Wikipedia. It showed that the findings from trust game experiments can be extended to real-world settings. The condition is that we need to tailor the trust model for each particular context.

In the next section, we will discuss several important studies that relate to our three research questions.

1.4 Related Work

In this section, we review several important state-of-the-art studies which are related to our three research questions.

- How do previous research works study the influence of different information on user behavior?
- What trust models have been presented for repeated trust game and Wikipedia?
- How do previous research studies predict signs of future links in signed directed networks?

1.4.1 Studying user trust under different circumstances with trust game

To the best of our knowledge, there is no previous work that studies the influence of trust score on user behavior in collaborative contexts. There is no evidence that users will listen and react to trust score. However, as we discussed above, the popular mechanisms to let users assess the trust level of their partners are reputation score, avatar, nick-name and review. The effect of these mechanisms on user has been studied using lab-control experiments such as trust game [Riegelsberger, M Angela Sasse, et al., 2003].

Existing research studies analyzed the influence of these mechanisms on user behavior to verify whether these mechanisms could change the behavior of users or not. [Yao and Darwen, 1999; Gary E Bolton et al., 2002] studied the influence of reputation score on user behavior in repeated games and suggested that introducing the reputation score could reduce the deviation of users. [Charness and Gneezy, 2008] studied the effect of revealing name in dictator game and suggested that if the users have to reveal a part of their names, they will share more. [Karlan, 2005] studied the effect of using nickname in trust game and concluded a similar observation. [Bente, Dratsch, Rehbach, et al., 2014; Yuksel et al., 2017] analyzed the effect of avatars on user behavior in a lab-control experiments. [Park et al., 2007; Duan et al., 2008; J. Lee et al., 2008] analyzed the influence of reviews from other users on the decision of a new user in e-commerce systems and stated that the influence of user reviews on buying decision is not very clear. Generally speaking, existing studies showed that while some mechanisms have positive impacts on user behavior, the influence of some other mechanisms are not clear.

In this thesis, we will study the influence of trust score on user behavior using repeated trust games. The experimental results could give us some more insights about how users will react to trust score in collaborative systems. As we discussed in Section 1.3.2, the effects we observed in repeated trust games could be found in real-world collaborative systems.

1.4.2 Calculating trust score

Several studies claimed that trust depends on the context [J. Tang, Gao, et al., 2012; Granatyr et al., 2015; Pinyol and Sabater-Mir, 2013; Sherchan et al., 2013; Rosaci et al., 2012], i.e. different environments require different different trust models. However, the existing reputation/trust models rely on a common principle that we discuss in Section 1.4.2.1. Then we discuss about model evaluation, i.e. how can we claim that a model is better than another one. Finally we review important state-of-the-art trust calculation for two case studies: repeated trust game and Wikipedia.

1.4.2.1 General Principle

Different environments require different trust methods. In this thesis we focus on Wikipedia and repeated trust game. However, the computational trust models rely on a common principle that trust is built based on past behavior of partners as we discussed in Section 1.1.2 and in general, a partner who behave well in the past is expected to behave well in the future. The idea is the core idea of many reputation and trust models in different settings [S. Ba and Paul A Pavlou, 2002; Weisberg et al., 2011; Xiong and L. Liu, 2004; B. Thomas Adler and Alfaro, 2007; Cho et al., 2015]. The problem is how to define the term "good behavior" in different contexts.

1.4.2.2 Evaluation of Trust Models

We discuss about evaluation method of trust models, or how can we claim a trust model is better than another model.

A trust model will take as input a pair of users that are *trustor* and *trustee* and returns a numerical value which is the *computed* trust level from the trustor to the trustee. The output value can be normalized into range $[0, 1]$ so we suppose that the trust value of two users is a number between 0 and 1 inclusive. It is easy to define an arbitrary number of trust models. For instance, we can return a random number as a trust value.

According to [Malaga, 2001], reputation (and by inference trust score) comprises a prediction about future behavior. For instance, if Alice has a high score, we could expect to observe a good

behavior from her in the future. If she fails to do so, the score assigned to her is wrong. Hence to compare the trust models, we compare their predictive power.

As we discussed in Section 1.1.2, trust from Alice to Bob reflects the expectation of Alice on the future behavior of Bob. It means, in a collaborative system if Alice trusts Bob more than Carol, Alice expects that in the future Bob will contribute more than Carol to the sharing work. The computed trust values can be considered as *advice* to a user about what trust level she could assign to their partners. Therefore, a good computational trust model should produce good advice, i.e. the future behavior of a partner matches with the previous computed trust score of this partner.

We consider an example. Alice has two partners Bob and Carol. Two trust models are proposed. The first model computed the trust scores from Alice to Bob and Carol and claimed that the trust score from Alice to Bob is higher than the trust score from Alice to Carol. The second model suggested an opposite view. Alice checked and realized that in fact, Bob collaborated while Carol deviated. Alice could claim that the first trust model is a better model and she should follow its suggestion in the future.

We considered a trust model as good if we can predict the future behavior of partners based on the computed trust scores. As we discussed in Section 1.1.2, if a future malicious partner did not deviate in the past, there is no way to detect this partner. Hence, there is no perfect predicting model and we have to accept some misleading predictions. However, using real-world datasets we can evaluate and compare our computational trust model with other baseline models.

1.4.2.3 Calculating trust in repeated trust game

Besides studying collaborative behavior, trust game is also an important research tool to study human trust under different contexts. The exchanging amounts among users in trust game reflect their trust on partners [Brülhart and Usunier, 2012]. Generally speaking, if Alice sends more to Bob than Carol, we could claim that Alice trusts Bob more than Carol. However, the problem of designing a trust calculation method in trust game has not been studied comprehensively.

The most popular trust calculation in repeated trust game is the averaging method [Glaeser et al., 2000; Burks et al., 2003; Karlan, 2005; Johnson and Mislin, 2011; Dubois et al., 2012; Murnighan and L. Wang, 2016; Butler et al., 2016]. According to the averaging method, trust score of a partner to a user is simply the average value of sending amount from this partner to this user in the past.

The advantage of the averaging method is that it is very straightforward. Non-technical users can easily understand the method. In fact the averaging method is widely used in many real-world systems today [Jøsang, Ismail, et al., 2007; Tavakolifard and Almeroth, 2012].

However, the averaging method is not able to cope with fluctuations and cheating behavior, such as a malicious partner who collaborates in the beginning to gain the trust of users before deviates. It also does not take into account the time information, i.e. the averaging method consider a recent action as same weight as an action since a long time [Jøsang, Ismail, et al., 2007].

1.4.2.4 Calculating trust of Wikipedians

In this thesis, we aim to compute the trust score of Wikipedians based on the quality of their contributions. Quality of user contributions relies on quality of Wikipedia articles. The trust score can assist users in assessing trustworthiness of their partners, such as in Wikipedia Request for Adminship (RfA) process [Burke and R. E. Kraut, 2008].

Assessing the quality of Wikipedia articles We quickly review several state-of-the-art studies in automatically assessing quality of Wikipedia articles in literature.

We can roughly divide the existing approaches into two families: editor-based approaches and article-based approaches [Warncke-Wang, Cosley, et al., 2013]. The editor-based approaches rely on the idea that a high quality document is written by good editors [M. Hu et al., 2007], therefore we can analyze the editors of an article to determine its quality [Betancourt et al., 2016]. On the other hand, the article-based approaches focus on the content of an article to determine its quality. For instance, [Blumenstock, 2008] suggested that a longer article tends to have higher quality than a short one. Following studies introduced more features into the model [Dalip, Goncalves, et al., 2009; Dalip, Lima, et al., 2014].

As of this writing, state-of-the-art in assessing quality of Wikipedia articles belongs to the work of [Warncke-Wang, Ayukaev, et al., 2015], wherein the authors defined eleven features to describe Wikipedia articles such as length of articles or number of images the article has. The set of features then is fed to a random forest model for classification. The model has been used by Wikimedia ORES service [Halfaker and Taraborelli, 2015]. However, the performance of these quality assessment methods is not very high: the state-of-the-art model achieves the accuracy score of only 62% in classifying six quality categories of 30,000 English Wikipedia articles [Wikimedia, 2016].

The set of eleven features describes Wikipedia articles intensively. However they do not consider how the articles have been written. Studies suggested that writing style does matter in measuring quality of Wikipedia articles [Lipka and B. Stein, 2010]. Furthermore, [Warncke-Wang, Ayukaev, et al., 2015] focused on *accuracy* score only, which does not cover all aspects of performance evaluation of a machine learning algorithm. Studies suggested that another metric which is more robust than *accuracy*, such as AUC, should be used [Huang and Ling, 2005; Japkowicz and Shah, 2011].

Existing approaches rely on traditional machine learning with manual feature engineering. Therefore, a new feature set is required for each language of Wikipedia. It makes the existing approaches difficult to generalize.

Assessing trust of Wikipedians In this section we review several approaches on assigning trust levels to Wikipedians. In fact, existing studies designed reputation models rather than trust models for Wikipedia editors.

WikiTrust [B. Thomas Adler, Chatterjee, et al., 2008] is a project to assess trust level of information presented on Wikipedia. Based on the trust level of information, we can assess the reputation level of Wikipedians [Javanmardi, Ganjisaffar, et al., 2009]. Unfortunately, at the time of writing the WikiTrust service is not available [WikiTrust, 2017].

Several studies focused on measuring user contribution to Wkipedia quantitatively, i.e. how much a user contributed to Wikipedia regardless the quality of the contribution [R. Agrawal and Alfaro, 2016]. The official metric which is being used by Wikipedia is the number of edits [Wikipedia, 2017f]. [B. Thomas Adler and Alfaro, 2007] proposed to use *edit longevity*, i.e. how long does a piece of text survive on Wikipedia, to measure the quality of text and then measure the reputation of the author of this text. The idea of the authors is that, if a text survives in a longer period of time, the text has higher quality. The authors compared their approach with the naive approach as counting the number of edit. The issue with the approach is that, in fact the edit longevity of a particular text can be determined exactly after this text is removed. In other words, this approach is not applicable for new content because there is no information on the survival time. For instance, if a user has recently inserted a new information to Wikipedia,

it is not possible to measure the longevity of this information: we have to wait at least a certain period of time to see if this text can survive or not.

Studies suggested that, in order to calculate the contribution value of a user, quality is more important the quantity [Alfaro et al., 2011; B Thomas Adler, 2012]. Nonetheless, no existing model takes into account the quality of Wikipedia articles. In this thesis, we propose a trust model that considers the quality of articles to calculate trust scores of Wikipedians.

1.4.3 Predicting trust relationship

The collaborative systems wherein users declare their trust/distrust explicitly on other users can be well described by a graph. The vertices of the graph represent users whose relations between them can be represented as positive or negative links [J. Tang, Y. Chang, et al., 2016]. These networks are called *signed directed networks* [Song and Meyer, 2015]. In these networks, a positive link can be interpreted as a trust relation from a user to another while a negative link can be interpreted as a distrust relation [DuBois et al., 2011; Ye et al., 2013].

By modelling a system as a signed directed graph, the task of predicting trust/distrust relations between users now turn to be the task of predicting positive/negative sign of future links that will be added to the network.

The link-sign prediction can recommend users to trust or not to trust partners who has never interacted with the user. As we discussed above, the size of modern collaborative systems are huge and it is difficult for a user to manually analyze the trustworthiness of partners that she has not known before.

Research studies claimed that we can infer unknown edge status easily by using personal information [J. Tang, Gao, et al., 2012; Ye et al., 2013] such as sociological information or personal trading history. In fact, many early trust prediction models exist relying on the similarity of two users [Ning et al., 2015], which in turn require access to personal information of users. However, due to the increasing concern of privacy on the Internet [X. Chen and Shi, 2009; Trottier, 2016], this information is usually neither available nor reliable. In order to reduce privacy concern, we aim to use *graph-based* algorithms [Jiang et al., 2016].

We notate a signed directed graph as $G =< V, E >$ where V is the set of vertices which represent users, and E is the set of links, or edges, which represent relationships between users [Leskovec et al., 2010a; J. Tang, Gao, et al., 2012][5]. Link-sign predictors assume existence of a graph where signs of all edges are known, except for an edge from node u to node v, denoted $u \to v$. The task is to predict the sign of $u \to v$, denoted $s(u, v)$ by using the information provided by the rest of the network [Leskovec et al., 2010a].

The input of graph-based algorithms is the graph of connections between users. We display an example of user connection graph with positive/negative directed links in Figure 1.3. A graph-based algorithm takes this graph as an input to inference the missing sign (from Alice to Carol in this example) based on the information of other edges.

For graph-based algorithms, there is no distinction between vertices because there is no personal information such as gender or personal preferences provided. The only reliable information is the topology of the graph. Therefore the graph-based algorithms preserve privacy.

One of the first studies in link-sign prediction is [Guha et al., 2004]. Firstly the authors represented a graph as a user relation matrix, which is still the most popular data representation in the field [J. Tang, Y. Chang, et al., 2016]. In user relation matrix, each cell represents a link from a user (row) to another user (column). The corresponding relation matrix of the graph

[5] In this thesis, we used the terms *graph* and *network* refer a same concept. Similarly, the terms *edge*, *edge* and *link* are used interchangeably. We also do not distinguish two terms *vertex* and *node*.

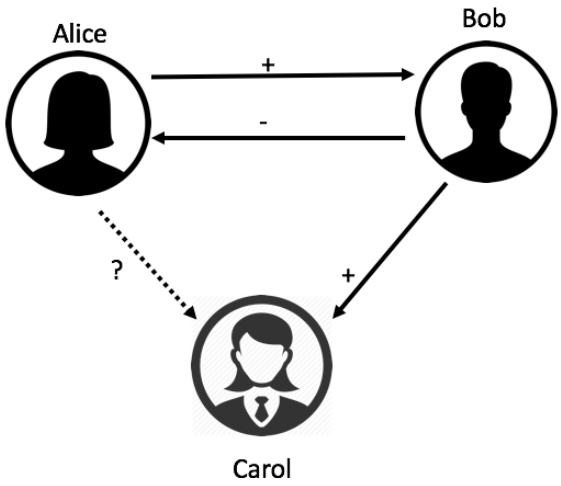


Figure 1.3: A social graph with signed directed connections.

displayed in Figure 1.3 is presented in Table 1.1. After that, the authors applied several well-known long-history rules such as "friend of friend is friend" and "friend of enemy is enemy" in term of matrix operations to predict missing signs.

	Alice	Bob	Carol
Alice		+	?
Bob	-		+
Carol			

Table 1.1: User relation matrix

Many link-sign prediction studies [Song and Meyer, 2015; J. Tang, Y. Chang, et al., 2016] rely on two sociological rules that are *structural balance theory* [Easley and Kleinberg, 2010, Chapter 5] and *social status theory* [Leskovec et al., 2010a].

Despite of the success of structural balance theory and social status theory, these rules are not very suitable for using in sparse networks. Furthermore, these rules ask for fully observed networks. [Song and Meyer, 2015] presented a Bayesian inference to predict link-sign prediction in partial observed networks.

The main problem with existing approaches is that they all work on static graphs, i.e. they take a snapshot of a social network and analyze the social network at this given point of time. However, modern social networks are very dynamic and the topology of the networks change every second. It is not realistic to train everything from scratch whenever a network changes. The challenge is to design a link-sign prediction method that can adapt to new information, such as the change of the network topology, while using the previous information.

In fact, the graph-based link-sign predictions can be applied to any signed directed graphs. In this thesis we focus on one application of these algorithms that is to predict trust/distrust relations of users in collaborative systems.

In this section, we discussed and highlighted important studies related to our research questions. In next section, we will briefly summarize our contributions for each research question.

1.5 Contributions

In this section we briefly describe our contributions for the research questions we presented in Section 1.2.

1.5.1 Studying influence of trust score on user behavior

In order to assist users in assessing the trust level of their partners, the popular mechanisms are identity, avatar, reputation score and review. As we discussed in Section 1.3.2, previous studies analyzed the effect of these mechanisms on user behavior. However, the effect of trust score on user behavior has not been yet analyzed.

Using repeated trust game, we tested the effect of trust score. Our experimental settings follow previous studies [Colombo and Merzoni, 2006; Avner Ben-Ner and Putterman, 2009; Dubois et al., 2012; Buntain and Golbeck, 2015] to test the effect of a new information given to users.

We recruited six groups of five participants to play repeated trust game under four conditions: (i) no information is displayed, (ii) partner identity is displayed, (iii) trust score of partner is displayed, and (iv) both identity and trust score of partner are displayed. We reviewed literature and stated the weakness of reputation schemes against attacks from third-party. We addressed the scaling problem of identity and avatar in large-scale collaboration. We analyzed the user behavior and claimed that: (i) introducing trust score improve the collaboration between users, (ii) trust score has a comparable effect with nick-name with no additive effect, (iii) users *trust* and follow the guidance of trust score, and (iv) trust score has a better predictive power than reputation score.

The results suggest that trust score could be deployed in real-world collaborative systems to assist users in assessing trustworthiness of their partners.

1.5.2 Designing trust calculation methods

Based on trust game experimental results, we claimed that trust models could be deployed to encourage and guide users in collaboration. The next question is how do we calculate trust score between a pair of users, given their interaction history. In this section we quickly describe our proposed trust model for repeated trust game and Wikipedia.

1.5.2.1 Computational Trust Model for Repeated Trust Games

Studies [Sapienza et al., 2013; Brülhart and Usunier, 2012] suggested that sending amounts between players in trust game represent their trust on each other . However, it is not clear how to build up a trust model given history of sending amounts between two users.

We present a novel computational trust score for repeated trust games that deals with fluctuating behavior. The main idea is to compute trust as a function of the amount exchanged in an interaction and accumulate it over several interactions. To deal with misbehavior, we record over time the change pattern in behavior. When the accumulative change factor exceeds a threshold, i.e. the partner fluctuated too much over time, this user trust score is decremented.

We validated the trust model against: (i) simulated data generated based on the meta-analysis [Johnson and Mislin, 2011], (ii) human rating [Keser, 2003], and (iii) real experimental data from trust game experiments, organized by ourselves and external trust game dataset provided by other studies [Bravo et al., 2012; Dubois et al., 2012].

To the best of our knowledge, it is the first presented computational trust model for repeated trust game. As trust game is an important tool in studying human behavior, we considered the work as a contribution not only for studying effect of trust score on user behavior but also for game theory research field.

The work is published in [Q. Dang and C. Ignat, 2016a].

1.5.2.2 Trust Calculation for Wikipedia Authors

In order to design a quality-based trust calculation for Wikipedia, we need to design first a method to automatically assess the quality of Wikipedia articles. We presented three different approaches to measure the quality of Wikipedia articles. Then we presented a method to calculate user trust score based on their contribution history.

Measuring quality of Wikipedia articles We presented three different approaches to measure quality of Wikipedia articles described in what follows. Each approach has its own pros and cons that will be discussed later.

Manual feature engineering approach As we discussed above, the state-of-the-art approach in assessing quality of Wikipedia articles is the approach presented by [Warncke-Wang, Ayukaev, et al., 2015] wherein the authors used a model of 11 features such as the length of article, the number of image, etc. to predict the quality of articles.

We improved the state-of-the-art by introducing nine additional features which are readability scores into the feature set of [Warncke-Wang, Ayukaev, et al., 2015]. The extracted features are fed into a random forest model. We performed different evaluation methods to test the performance of the new model. The experiments on English Wikipedia dataset claimed that the new model achieved an *accuracy* score of 64% and AUC score of 0.91 compared to an accuracy of 58% and AUC of 0.87 of state-of-the-art. Furthermore, statistic test confirmed that the performance difference between two models is significant.

The work has is in [Q. Dang and C. Ignat, 2016b]. We will refer this model as random forest based approach in this thesis.

Deep learning approaches Traditional machine learning algorithms relies on carefully selected features [Guyon and Elisseeff, 2003]. The feature selection process is mostly based on expertise of researchers. There is not yet a way to extract the best features from a given dataset [Stanczyk and Jain, 2015]. In practice, feature selection is done by listing as many features as possible then evaluating them to eliminate non-relevant features [Stanczyk, 2015]. However, this approach cannot find missing features. Different Wikipedia language require different feature set [Wikimedia, 2016].

We present two novel approaches on assessing quality of Wikipedia articles that do not require manual feature engineering. These approaches can be used in any language of Wikipedia.

The first approach uses $Doc2Vec$ [Le and Mikolov, 2014] to convert Wikipedia articles into numerical vectors then feed these vectors into Deep Neural Networks (DNN) for training and predicting [Goodfellow et al., 2016, Chapter 6]. The approach achieves the *accuracy* score of 55%, not far from [Warncke-Wang, Ayukaev, et al., 2015], but is much faster in term of development time, i.e. a beginner can implement the approach in few days, while it took several years for researchers team to come up with the approach of [Warncke-Wang,

Ayukaev, et al., 2015] or [Wikimedia, 2016]. The work is published in [Q. Dang and C. Ignat, 2016d]

In the second approach, instead of using $Doc2Vec$ which is very expensive in term of computation, we used Recurrent Neural Networks (RNN) [Rumerhart et al., 1986] with Long-Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] for building an end-to-end learning method. The approach achieves higher *accuracy* and AUC scores compared to the random forest based approach. However, the running time is much longer: the RNN-LSTM approach takes several days for training and several hours for testing using a powerful server while the random forest based approach takes several minutes for training and several seconds for testing on the same dataset using a Macbook Pro Mid 2014. The work is published in [Q. Dang and C. Ignat, 2017].

The deep learning approaches achieve higher *accuracy* and AUC and are available for any language without human intervention, but the cost is a longer running time. Therefore, the selection of solution depends on the application requirements and computational resource.

Calculating trust score of Wikipedians Consider a scenario: a Wikipedia editor wants to decide which partner she should collaborate with to write a new Wikipedia article. She wants to see who is the most effective partner among ones collaborated with her in the past. Unfortunately, she realized that she has collaborated with so many different partners and now she is not able to remember who is who, let alone to determine their collaboration quality.

We applied the computational trust model we presented in Section 1.5.2.1 for Wikipedians. Our method will scan through the user's history and calculate trust score of each partner based on the quality of collaborative works. We validated our algorithm in real-world Wikipedia dataset. The experimental results suggest that, given the quality information of collaborative articles we can better assign trust score to users and predict their future contributions than other averaging baseline methods.

1.5.3 Predicting trust relationship

In the previous section we described how we calculate the trust score between two users if they interacted. However in collaborative systems there are pairs of users who did not interact but one needs to assess the trust level of the other. In this section we describe our contribution in predicting future trust/distrust relationship of users. We proposed a new link-sign prediction for this task.

Several link-sign prediction algorithms have been presented in recent years as we discussed roughly in Section 1.4.3. Existing methods mostly rely on traditional machine learning techniques which require manual feature engineering and require fully observed networks which are usually not available in practice [Song and Meyer, 2015]. Furthermore, these algorithms need to be trained from scratch if the network changes.

We presented an approach that combines Random Walk, Doc2Vec [Le and Mikolov, 2014] and Recurrent Neural Network (RNN) [Goodfellow et al., 2016, Chapter 10] for link-sign prediction in dynamic networks. Our contributions are:

- Our algorithm requires only local information.
- Our algorithm can be trained incrementally, i.e. if the network changes we only need to update the new information to the trained model without learning everything from scratch.

- Our algorithm outperforms state-of-the-art in evaluation using real-world datasets.

Therefore, our algorithm is more suitable for dynamic networks, wherein nodes and links are established and removed frequently.

As we discussed before, the link-sign prediction can be used as trust/distrust prediction in collaborative systems. We can predict and assist users in assessing the trustworthiness of partners that they did not interact with before.

1.6 Outline

The thesis is organized as follows.

Chapter 2 presents our study for the first research question: "should we introduce trust score to users?". We analyze the weaknesses of popular techniques such as identity, avatar and reputation score and how using trust score can resolve these problems. Then we describe our experimental design and analyze the influence of trust score on user behavior.

Chapter 3 presents our study on calculating trust score, which is the content of the second research question. We present two trust calculation methods for two environments: repeated trust game and Wikipedia. Each trust calculation method is validated against real-world datasets.

Chapter 4 presents our algorithm for link-sign prediction in dynamic large-scale signed directed networks. The algorithm is a combination of Random Walk, Doc2Vec and RNN. The algorithm is validated on popular real-world datasets and it achieves better performance in term of accuracy score and F1-score compared to state-of-the-art.

Chapter 5 concludes the thesis and draws some potential future research ideas.

Chapter 2

Influence of Trust Score on User Behavior: A Trust Game Experiment

> You have to learn the rules of the game. And then you have to play better than anyone else.
>
> — Albert Einstein

Contents

In this chapter, we address the first research question: "Should we introduce trust score to users in collaborative environments?". We divide the research question to following problems.

1. Does the availability of trust score have effect on user behavior in collaboration?
2. Do users follow the guidance of trust score?

In this chapter, we address the above problems. First we will analyze the problems of existing popular mechanisms to help users in assessing trust level of their partners, which are avatar, nick-name, review and reputation score. We analyze how can trust score resolve these problems. We distinguish between trust and reputation because these two concepts are easily be confused. Then we will describe our experimental design to study the influence of trust score on user behavior in trust game. After that we present the experimental results and conclusions. Lastly we will discuss the external validity of the experimental findings in real-world systems.

2.1 Methodology

In a large-scale collaboration, a user interacts with a large number of partners. It is helpful for a user to assess the trust level of each partner, so the user can decide to collaborate with which partner. We propose to deploy a trust model to assist user in trust assessment. The trust model takes into account the interaction between a user and a partner to calculate the trust score of the user on the partner. The trust score will be displayed to the user as the suggestion of the model about the trust level of the partner. To the best of our knowledge, there is no real-world system that integrated a trust model. We do not know the effect of showing trust score to user behavior. In this chapter, we validate the influence of trust score on user behavior in trust game environment. We address two questions relate to the choice of using trust game:

1. Existing techniques to assist users in assessing trust level of their partners in large-scale collaboration are available. The popular ones are reputation score, nick name, review and avatar. Why does the trust score should be used?
2. Why did we use trust game instead of real-world applications as the experimental environment?

2.1.1 Review on existing techniques

Several techniques are used today on the Internet to assist users in assessing the trustworthiness of their partners. The most popular ones are nick name, avatar, review and reputation score.

We review these techniques in the following sections. We show that they have several critical shortcomings making them not be suitable in large-scale collaboration. We will also argue that trust score can resolve these problems.

2.1.1.1 Nick name and avatar

Nick-name is a string that is assigned by the system or chosen by users to represent their identities. Nick-name is a widely-used mechanism today to allow users to identify their partners.

We note that in this thesis, we used the term *nick-name* and *ID* interchangeably. Thanks to the identification, users can recall their history with the partners. They can assess the trust level of the partners based on her experience [Bhargav-Spantzel et al., 2007]. Studies [Bays, 1998] suggested that Internet users maintain their *faces*, or their images, through their nick-names so other users can recognize and accept them.

The nick-name aligns with our experience in daily life: we identify and remember other people by their names. However, in large-scale online contexts, a nick-name system has several shortcoming. Studies showed that it is very difficult for human to remember non-sense strings, such as *pan2216771887bOz* [Dix, 2009]. This kind of string is being used on the Internet as nick-name. On the other hand, the nick-name can be "faked", as a malicious user could create a new user with the nick-name as *pan2216771887bOz* which is not easy to recognize by ordinary users - the same idea is applied for "fake" famous brands, such as *Panasonic* or *Panansonic*. Moreover, users can change their nick-name to distinguish the bad experience of other users on the old nick-name.

More concerning is that, it is difficult for typical users to remember nick-name of every partners in large-scale collaboration. Psychological research has established the persisting response time penalties of increasing the size and interconnectedness of declarative content such as ID [Anderson and Reder, 1999]. As a result, increasingly large, dense networks with rarely accessed nodes, such as those made possible by internet collaboration, pose retrieval problems, and hence access to the knowledge that supports effective collaboration. [Riegelsberger, Martina Angela Sasse, et al., 2005] specifically notes the overhead associated with the maintenance of partner-specific information.

Avatar is another mechanism to assist users in assessing trustworthiness. Avatar is a photo chosen by the system or by the user to represent the user. The core idea of using avatar is similar to using nick name: a user look to an avatar of a partner to identify the partner and recall their experience with this partner. Studies suggested that users can remember images better than text [Dhamija and Perrig, 2000] so using avatar can enhance the quality of user suggestion.

Studies suggested user preferences on avatar. For instance, [Bente, Dratsch, Kaspar, et al., 2014] suggested that a portrait avatar will gain more trust from partners than a random image. Furthermore, [Bente, Rüggenberg, et al., 2008; Yuksel et al., 2017] suggested that in the same context participants tend to trust partners with more beautiful face avatars.

Avatar shares the same limitation with nick-name system. In large-scale collaboration it is difficult for users to remember avatar of every partners. It is easy to fake an avatar, i.e. one can use a photo of another person as their avatar. Furthermore, a honest user might lose their trust relationship she built with other users when she changes her avatar.

We conclude that both ID and avatar systems do not scale well to large-scale collaborations.

2.1.1.2 Reputation score

Reputation score is another method to measure the trustworthiness of users. Using a single score such as a reputation score can overcome the limitation of nick-name or avatar, because users do not need to remember anything. When Alice observes a score of Bob she can decide her next activity with Bob.

Trust and *reputation* are used sometimes interchangeably in literature [Vu et al., 2010; Sun and Ku, 2014; Pecori, 2016]. They are close but not the same concepts [Fetchenhauer and Dunning, 2009]. *Reputation* is a collective opinion from community to a particular user, while *trust* is a personal opinion from a user to another user. As [Abdul-Rahman and Hailes, 1997] stated, trust is an *one-to-one* relation, while reputation is a *many-to-one* relation. Reputation

value of a community on a user is a *global* value, i.e. everyone from this community will see a same reputation score of the user. On the other hand trust of a user on a partner is a *personal* value and different from partners [Hoelz and Ralha, 2015; Ert et al., 2016], i.e. different users will have different trust score on a same partner.

The scoring systems used widely on the Internet today are reputation systems, not trust systems [Resnick, Kuwabara, et al., 2000]. Examples include Amazon rating system (Figure 2.1). The score (4.5 stars in this case) is reputation score because it is *global*, calculated by averaging all rating score from all buyers. Every buyer will see the same score when they look to the profile of the seller.

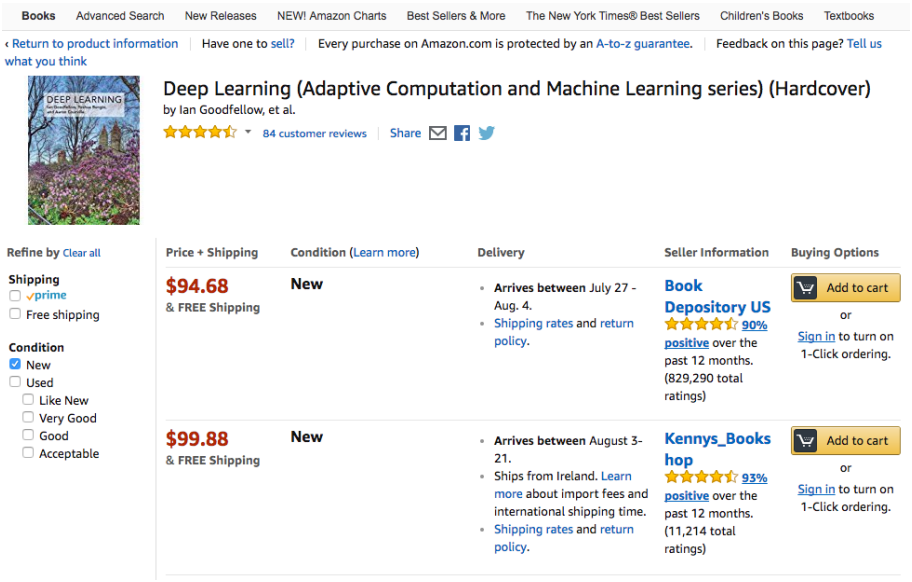


Figure 2.1: A screenshot from Amazon: the website shows the reputation score, which is average rating score.

Studies [Yao and Darwen, 1999] proved the effectiveness of reputation score in collaborative games. In general higher reputation leads to higher trust [Doney and Cannon, 1997]. However the covariance relationship between reputation and trust might be disrupted by malicious behavior. One of the most popular attacks to reputation systems which is called *discrimination* [Sun and Ku, 2014]. It occurs when a service provider offer bad services to only a proportion of users. By doing that, the service provider can gain both revenue and reputation score.

Beside discrimination attack strategy, several other attack strategies to reputation schemes exist [Hoffman et al., 2009]. The vulnerability of reputation score is not as obvious as nick name and avatar, and reputation score is a close concept with trust score. We review these attacks in more details in Section 2.1.1.3. Designing defense techniques for some of these attacks are still open questions today [Sun and Ku, 2014] but as we will see in Section 2.1.1.3, trust score is robust against these attacks.

Another problem with reputation schemes is that they require a central server to operate [Aberer and Despotovic, 2001]. The schemes might not scale well in large-scale collaboration.

Furthermore, the reputation scores lack personalization. According to [Y. Wang and Vassileva, 2007] a personalized score is required in the presence of subjective factors, i.e. user needs or interests. A *personal trust* scoring system can overcome such limitations. A trust score is ideally calculated and attached to a participant. Because the trust score reflects personal expe-

rience between pairs of users, the playbook attack is not possible. A user can compute the trust score of a partner locally without querying information from third parties. Crucially, with trust scoring, participants do not need to recall anything. Continued, context sensitive interaction therefore proceeds without the cognitive demand.

In the next section we describe in details attacks on reputation score. We show that there is not yet effective solutions for some of these attacks.

2.1.1.3 Attacks on reputation score

Due to the fact that reputation score is widely used on the Internet today and it is a very close concept to trust score, it is important to address several popular critical threats to reputation score [Hoffman et al., 2009; Sun and Ku, 2014].

We divide attacks on reputation score into two groups *malicious individual* and *malicious collective*. Malicious individual means that even a single malicious user can start an attack, while malicious collective means that in order to establish an attack two or more malicious users are required.

Malicious individual [Sun and Ku, 2014] listed three attack types that belong to this group. They are *providing inauthentic services* [Mármol and Pérez, 2009], *Sybil attack* [Douceur, 2002] and *playbook* [Y. Wang and Vassileva, 2007].

Providing inauthentic services. This attack could be considered as a *naive* attack model, when the attacker did not provide a service as advertised before. This attack type is easy to deal with [Sun and Ku, 2014], such as blocking and reporting the malicious partner to the system or to other users.

Sybil attack. Sybil attack is defined as, the attacker continuously create new identity to offer bad services. There is not yet a very good method to handle this kind of attack. A popular suggestion is to create an entry barrier to the system to make it difficult for attackers to create new identities. On the other hand the system will lose its availability to honest users [Dinger and Hartenstein, 2006].

Playbooks. A playbook can be defined as a *strategy* of an agent to maximize their own profit [Y. Wang and Vassileva, 2007]. A typical playbook is that an attacker plays honestly in the beginning then suddenly deviate, and then the attacker can even plays honestly again to recover the reputation level. In some systems, this kind of action is not punished so the reputation can be recovered easily, and in fact "most of the trust and reputation mechanisms prove useless for malicious agents to behave in this way" [Sun and Ku, 2014].

Together with Sybil attack, a possible defense technique is to increase reputation or trust score slowly in the beginning and punish immediately if the cheating behavior is detected.

Malicious collectives A malicious collective can be formed when multiple attacker establish a collation to increase reputation score of each other [Mármol and Pérez, 2009; Jøsang and Golbeck, 2009].

Malicious spies. Many reputation schemes [Yamamoto et al., 2004; Allahbakhsh et al., 2012] are influenced by the idea of PageRank [Page et al., 1999], i.e. a user is good if she is rated as good by other good users. Therefore, a malicious spy can be created: a malicious spy is an agent who always behave honestly so she earns a high reputation, then she can

give high rating values to other malicious users. Because a malicious spy owns a high reputation score, her rating value has a high weight and can improve reputation scores for other malicious users.

Self-Promoting. *Self-Promoting* is defined as attackers try to increase their own reputation score [Hoffman et al., 2009]. There may be several attackers, or several accounts which belong to an attacker, give positive feed-backs and increase reputation score of each other.

White-Washing. *White-Washing* is related to Sybil attack: an attacker gave up their old account and create a new one to dismiss with bad experiences. However, these account exist longer than in the case of Sybil attack, so the attacker can even gain some reputation.

Slandering. *Slandering* is an opposite side of Self-Promoting attack. In Slandering attack, attackers try to decrease the reputation score of honest users by declaring negative feed-backs and giving low rating values [S. Ba and Paul A. Pavlou, 2002].

Discrimination. Discrimination attack occurs when an attacker cheat only a small proportion of her partners, while playing honestly to the remains.

The *discrimination* attack can be explained by a simple example. Suppose a service provider has 100 customers. For each customer, if the provider plays honestly she will earn a profit of 1, and if she cheats this customer, the profit will be 10. A customer who has been treated honestly will rate the score of 5, while customers who are cheated will give the rating score of 1. Now, if the service provider treated all customers honestly, she earns an average rating score of 5 and profit of 100. However, if she cheats 10% of customers, she earns an average rating score of 4.6 which is still high, but now she earns the profit of 190 - almost double.

If the number of partners is large enough, the attacker then can simply skip the partners she already cheated and continue with the others. Furthermore, the partners who received good services can introduce the service to their friends, increase the number of potential victims of the attacker. Even in the case of limited number of partners and these partners may know each others, the attacker can still offer her service for a long time before she is denied by all partners [Easley and Kleinberg, 2010, Chapter 21].

It's easy to prove that, in order to maintain the reputation score at the value of s, in each step the attacker can cheat $\frac{5-s}{4}$ of her partners. For instance, if the attacker wants to maintain the reputation score of 4, at each step she can cheat 25% of her partners.

Furthermore, reputation score relies on a very important assumption, named partner-independent behavior pattern, i.e. the reputation schemes assume that a user will perform the similar activities to all partners. The reason is, the reputation score of a user, say Alice, is global, so Bob and Carol observe a same reputation score of Alice and they could expect the similar actions from Alice. We will analyze the predictive power of reputation score in Section 2.1.3 and 2.4.3.

Defense techniques & trust score Up to now, designing an effective defense technique for malicious collective attacks is still a difficult problem [Sun and Ku, 2014]. Malicious collective attacks are based on the fact that the reputation score can be affected by third-party. Due to the nature of trust score which is a personal score and can not be affected by third-party, trust score is robust against malicious collective attacks.

Therefore, a trust model can deal with malicious individual attacks. We will discuss more about trust score calculation in Section 3.

However, we do not know yet the effect of showing trust score on user behavior in collaborative contexts. We will use trust game experiments to study this effect. In the next section we discuss about trust game.

2.1.2 Why trust game?

We deployed trust game experiments to study human collaboration when trust score is available. We present the most important descriptions about trust game here. More details information about trust game is available in Section A.1.

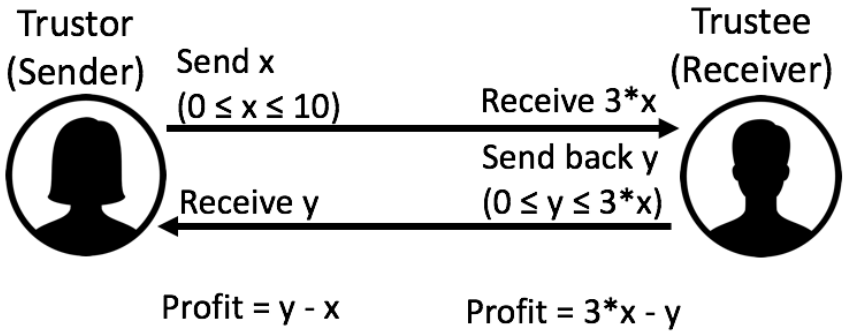


Figure 2.2: Trust game

Trust game is visualized in Figure 2.2. Trust game is a game between two players *sender* and *receiver*. The game is played in two turns. In the first turn, the sender selects an amount between 0 and 10 to send to the receiver. The amount will be tripled in the receiver's side. In the second turn, the receiver selects an amount between 0 and what she received to send back to the sender. In this turn, the amount the sender receives will not be tripled. The objective of each player is to maximize their profit.

The game pits joint payoff against individual payoff. Joint payoff is maximized if the sender returns 10 to the receiver, so the total profit is 20. However, assuming that users only seek to maximize their own profit, normative game theory predicts that the sender will send 0 and upon receiving any sum the receiver will send back 0. Any other amount other than 0 will reduce the receiver's profit. According to normative theory for the one-round trust game, the sender knows this fact, so at her turn, she should send 0 to the receiver because if she sends any greater amount, she must assume that she will not receive anything back [Camerer, 2003].

In fact, participants do not behave according to normative theory, but choose to maximize their joint profit. The trust game is therefore considered to be *cooperative* [Cesarini et al., 2008; Camera and Casari, 2009; Balliet and Van Lange, 2013], or said differently, increases in payoff reflect cooperation.

Trust game can be repeated, i.e. the game can be played in multiple rounds [Engle-Warnick and Robert L. Slonim, 2004]. The players can be switched, i.e. a player can play with different partners in different rounds [Dubois et al., 2012]. The roles of players can be switched also, i.e. a player can play as a sender in a round and as a receiver in another round [Burks et al., 2003].

We used trust game due to several reasons:

- The findings about human behavior in trust game can be extended to real-world systems [Baran et al., 2010]. Therefore, trust game is a valuable tool to study human behavior in collaborative contexts. We discuss in details about the external validation of trust game in Section 2.6.

  - Consider a collaborative editing system like ShareLatex. Alice started a document and wrote a part of this document. She realizes that it is very difficult to complete the document by herself, so she needs the help from a partner. However, if Alice shares the document with Bob, two cases can occur. Bob can contribute to the document by completing it, modifying the mistakes Alice made, adding information. The result is that Alice and Bob have a complete document, which is definitely more valuable than two incomplete parts written by Alice and Bob individually. However, Bob can deviate by adding false information, or removing good content written by Alice, or copy the parts of Alice to serve his own purpose, such as claim his copyright. In this case, Alice will lose her contribution and even more, such as time, effort and opportunity in term she missed the deadline to complete the document. The situation between Alice and Bob can be described as a trust game. The unknown information here is how can we define and measure the contribution, the profit and the loss of Alice and Bob.
  - Various studies have confirmed the external validity of experimental games in studying trust. We discuss this in more details in Section 2.6.

- It is very costly to implement a trust model in real-world systems such as Wikipedia.

- Deploying and controlling the experimental factors in trust game is easy. We can change one factor, such as displaying or hiding trust score, while keeping other factors constant.

- Studies proved that the exchanging amounts between users represent their trust on each others [Brülhart and Usunier, 2012], while trust representation in other contexts are rather challenging.

  - For instance, let's consider a case wherein a user A buy a product B of the manufacturer C from a seller D on a commercial website E likes Amazon or Wish. In this case, it is difficult to say A trust whom. In fact, A might trust the quality and functions of the particular product B, or trust C as a famous company, or trust D because some friends of A bought something else from D, or trust E because A knows that E can protect customers' rights well. The analysis in this case will be very complex.
  - On the other hand, in case of non-fulfillment [Riegelsberger, Martina Angela Sasse, et al., 2005], i.e. the expectation of the trustor fails, it might be difficult to determine that the fault belong whom. In the above example, if the user A receives a damaged stuff, it might be the fault of the seller or the post service. The analysis becomes over-complicated to analyze the factors that effect to human trust.

- Multiple studies on trust game have been presented. It allows us to reuse the existing data and compare with other experimental designs. In fact, experimental games and survey are two main measurement methods to study human trust [Dinesen and Bekkers, 2015].

2.1.3 Analyze predictive power of trust and reputation score

Above we reviewed the problems of reputation models and explained how a trust model may resolve these problems. Here we present a preliminary analysis of the predictive power of participants' future behavior in the trust game comparing trust and reputation scores.

As we discussed in Section 1.4.2.2, we evaluate the performance of a scoring method by its predictive power of future behavior of users. In this section we present the comparison between our trust model and a reputation model in predicting future behavior of users in two external datasets from [Dubois et al., 2012] and [Bravo et al., 2012].

We employed two external datasets from two repeated simple trust game experiments independently conducted by Dubois et al. [Dubois et al., 2012] in Montpelier, France and Bravo et al. [Bravo et al., 2012] in Bresica and Cuneo, Italy. The [Bravo et al., 2012] experiment involved 36 participants and contained five rounds. The [Dubois et al., 2012] experiment involved 108 participants and contained ten rounds. For computing trust scores we employed the trust function that we describe in Section 3.1.1, shown to reflect and predict user behavior in repeated trust game with resistance to fluctuating user behavior. As a reputation measure we used users average sending proportion up to the moment the reputation is computed, which is similar to many real-world reputation scoring methods [Jøsang, Ismail, et al., 2007; Tavakolifard and Almeroth, 2012].

We employed a regression analysis with the trust score computed by our trust function as a predictor and with observed sending proportion as the criterion. Starting with round 4 when the trust metric has stabilized, we predicted the send proportion of participants starting from round 4 by the trust score calculated after the previous round. We employed a similar regression analysis with the reputation score as a predictor and the sending proportion as the criterion. The results for the sender role are displayed in Table 2.1 and for the receiver role are displayed in Table 2.2. The corresponding t-values for the trust value assigned to each participant in predicting their future behavior are all significant for both senders and receivers, i.e. the trust score calculated by our trust function is predictive for external datasets. Adjusted R^2 are higher for trust values than for reputation in all cases except for round 7 and 8 for senders in the Dubois dataset. We recall that the trust model require less information (only information observed by the user) than the reputation model which requires full information from all users.

Dataset	**df**	**t-value for trust**	**Adj. R^2 for trust**	**t-value for reputation**	**Adj. R^2 for reputation**
Bravo dataset (round 4)	106	7.85***	0.36	3.05**	0.19
Bravo dataset (round 5)	106	10.0***	0.48	8.86***	0.42
Dubois dataset (round 4)	34	4.41***	0.35	3.24**	0.21
Dubois dataset (round 5)	34	4.51***	0.36	2.84**	0.17
Dubois dataset (round 6)	34	4.68***	0.37	4.26***	0.32
Dubois dataset (round 7)	*34*	*4.05****	*0.31*	*4.29****	*0.33*
Dubois dataset (round 8)	*34*	*4.15****	*0.32*	*4.83****	*0.39*
Dubois dataset (round 9)	34	4.25***	0.33	3.17**	0.21
Dubois dataset (round 10)	34	4.52***	0.36	2.36*	0.11

Table 2.1: Regression analysis of our trust function and reputation applied on external datasets for sender role.

While parity would be satisfactory, this preliminary analysis provides compelling evidence

Dataset	**df**	**t-value for trust**	**Adj. R^2 for trust**	**t-value for reputation**	**Adj. R^2 for reputation**
Bravo dataset (round 4)	93	4.72***	0.18	4.71***	0.18
Bravo dataset (round 5)	64	5.04***	0.27	4.61***	0.24
Dubois dataset (round 4)	30	3.84***	0.31	3.15**	0.22
Dubois dataset (round 5)	31	4.58***	0.35	2.95**	0.19
Dubois dataset (round 6)	31	6.06***	0.53	2.20*	0.11
Dubois dataset (round 7)	29	6.52***	0.58	2.93**	0.20
Dubois dataset (round 8)	30	6.69***	0.64	4.88***	0.42
Dubois dataset (round 9)	26	3.86***	0.34	1.59	0.05
Dubois dataset (round 10)	27	4.88***	0.45	4.38***	0.39

Table 2.2: Regression analysis of our trust function and reputation applied on external datasets for receiver role.

for the predictive power of trust scores. These are somewhat surprising findings given that none of these participants were aware of their partners. In fact, the trust function has more contextual parameters than reputation, accounting for partner, cumulative behavior over time, and punishment of misbehavior. In the next sections we present our research questions and experimental design for demonstrating the influence of trust scores on user cooperative behavior.

In Section 2.1.1.3 and Section 2.1.3, we showed by both logical arguments and real-world datasets analysis that a trust model can resolve some open problems of reputation models. However, while the effect of reputation model on user behavior has been studied [Yao and Darwen, 1999], there is yet an evidence that showing trust score will have any effect on the behavior of users. In the following section, we present our experimental design and analysis to study this question.

2.2 Experimental Design

In this section we present our experimental design using trust game to study the effect of showing trust score on user behavior. We displayed the terminology of the design in Table 2.3.

Term	Definition
Participant	A person who joined our experiment
Sender	The first mover in a round
Receiver	The second mover in a round
Round	A one-trial interaction between a sender and a receiver
Game	A game contains 25 rounds
Session	A set of six participants play all four games in a random order. A session contains 100 rounds in total
Group	A group of six participants who participate a same session

Table 2.3: Definition of terms used in the paper

		ID presented	
		FALSE	TRUE
Trust presented	FALSE	Simple Game: The trust game when participants have no information about partners	Identity Game: The trust game when participants have only nick names of partners
	TRUE	Score Game: The trust game when participants have only trust scores of partners	Combine Game: The trust game when participants are given both trust scores and nick names of partners

Table 2.4: Game descriptions

2.2.1 Participants

Participants were recruited through a public announcement. Five independent groups of six participants resulted in a total 30 of participants. Four of the five groups included one female participant, while the fifth group included two female participants. The ages of participants ranged from 19 to 45 with an average age of 28.5.

Typically researchers compensate participants using an exchange rate between virtual money in the experiment and real money, then pay the participants an amount based on how much they earned during the experiment. To assure continuing incentive throughout the session, each person who participated received a coupon of ten euros, but the person who earned most, i.e. who had the highest payoff among other people in the group, received an additional coupon of ten euros.

2.2.2 Task

The basic experimental task consisted of exactly 25 rounds of the trust game between a pair of players, during which a participant served as sender and receiver equally often. At the beginning of the first played game each user receives 10 money units. In each round, the sender moves first. She knows how much money she has, and must decide the amount she wants to send to the receiver. After that, the receiver receives a message indicating how much he has at the beginning of this round, how much he received from the sender, and how much he has after receiving. Then, the receiver decides how much she wants to return.

2.2.3 Independent Variables

We crossed the availability of nicknames and partner trust scores to create four different games as shown in Table 2.4. Nicknames, such as “Mr. Black" or “Mrs. Green", were assigned to participants, fixed during a game and varied between games. We do not describe the trust function here because the main objective of the chapter is to study the effect of trust score on user behavior, not trust function. Technical details about the trust calculation is presented in Section 3.1. Trust scores were always calculated for each participant in a pair, but only displayed according to experimental condition and only partner scores were available. The theoretical trust score value ranges from 0.0 to 1.0 inclusive, presented when available with two significant digits.

Participants started with the neutral value of 0.5 [Abbass et al., 2016].

We calculated user reputation_score as distinct from trust_score by averaging all previous sending proportion amounts of that user in both roles sender or receiver.

2.2.4 Design

The experimental conditions were organized as a split-plot factorial with group as a between subjects factor and Show-ID and Show-Trust as within subjects, such that each group of six participants participated in the set of four randomly ordered games. Show-ID and Show-Trust are two Boolean variables to indicate that identity and trust score of partners are presented or not. In each round, participants were paired randomly within their group. We ensured that in each game, a participant was paired with a particular other participant exactly five times. Within this pairing, two participants in a pair were assigned their roles randomly: one was the sender and one was the receiver.

2.2.5 Dependent Measures

Sending proportion by senders: the net amount the sender sends to the receiver over 10, which is the maximum amount the sender could send.
Sending proportion by receivers: the net amount the receiver sends back over the amount she received after being tripled. Other studies [Bornhorst et al., 2010; Bourgeois-Gironde and Corcos, 2011; Burks et al., 2003; Dubois et al., 2012; R. Slonim and Garbarino, 2008] also used sending proportion measures in order to normalize the sending behavior of receivers for comparison.

For example, sender A has sent 6 to receiver B, and B sent back 9 to A. In this round, the net sending amount of A and B are 6 and 9 respectively, the sending proportion of A is $6/10 = 0.6$ and the sending proportion of B is $9/18 = 0.5$.

Consistent with [Bornhorst et al., 2010; Burks et al., 2003; Dubois et al., 2012] for all analyses of receiver behavior, we eliminated the zero transaction between the sender and the receivers, i.e. the sender sends 0 and the receiver is obliged to send 0, for two reasons. First, receiver behavior is completely determined by the sender, so that the receiver's behavior is not informative. Moreover, in this case, the sending proportion for receiver (0 divided by 0) is not calculable.

For sender, there are exactly 375 data points in each game. For receiver, the number of data points vary between 250 (Simple Game) and 340 (Combine Game) due to the zero-transaction elimination.

Average sending proportion by senders: the average of sending proportions by each sender over all trials in the game. Taking an average distributes the effect of the zero transaction and also eliminates trial as a repeated factor in analysis.

Average sending proportion by receivers: the average sending proportion the receiver sends back to the sender over all trials in the game, without the zero transaction case from the sender.

Using average sending proportion for both sender and receiver will provide us 30 data points for each role. For receiver, the zero-transaction data is removed before calculating the means.

2.2.6 Procedure

All groups participated independently using z-Tree [Fischbacher, 2007] on our laboratory computers. At the beginning of each session, we asked all participants to read the instructions presenting the purpose of the experiment, a short description of the four games, the payment procedure and some example screenshots demonstrating the interaction of users with the zTree tool. The instructions informed participants that they will play the games in an arbitrary order. For each of the games participants were stated what partner information would be displayed during each interaction: for the Simple Game no information, for the Identity Game the partner identity in the form of a nickname, for the Score Game a partner trust score computed according to her behaviour in previous interactions (without any details about the metric) and for the Combine Game, the partner identity and trust score. Participants did not know the number of rounds they would play in each game. After confirming that they had read and understood the instructions and had no further questions, participants reviewed and signed an informed consent form prior to commencing the experiment. Participants were placed in different rooms to avoid any communication during the experiments. Each participant used a computer running our zTree application. All senders in the group finished their decision making process before proceeding to the next trial. Play then waited for every receiver to respond before starting a new round. This eliminated response time cues as an indication of player identity. No other means of communication or identification were available. Participants were informed of their cumulative earnings at each round.

It is possible to play with a negative balance but this never occurred.

The repeated measures design resulted in exactly 100 rounds across the four games. A session usually lasted two hours. At the end of the experiment participants filled out a questionnaire regarding general information such as university major and game preference.

In the next section we will present our analysis and findings of the user behavior collected from the experiments.

2.3 Results

In this section we present the results from our experiments. We divide the section into three subsections: behavior of senders, behavior of receivers, and considerations about our experimental design.

2.3.1 Sender Behavior

In this subsection we study the behavior of sender in responding to our manipulations. We show that both trust score and ID increase sending generosity with equivalent improvement and no combined effect. To examine cooperation, we analyze the 0 exchange condition and rule-out round effects as influential for all games except the Simple Game with no partner information. Finally, we demonstrate the dependence of performance on trust score metrics.

2.3.1.1 Omnibus ANOVA

We performed a basic ANOVA with Subject, Show-Trust and Show-ID as predictors. The analysis reveals an interaction, $F(1,29) = 19.36$, $p < 0.001$ as measured by average proportion

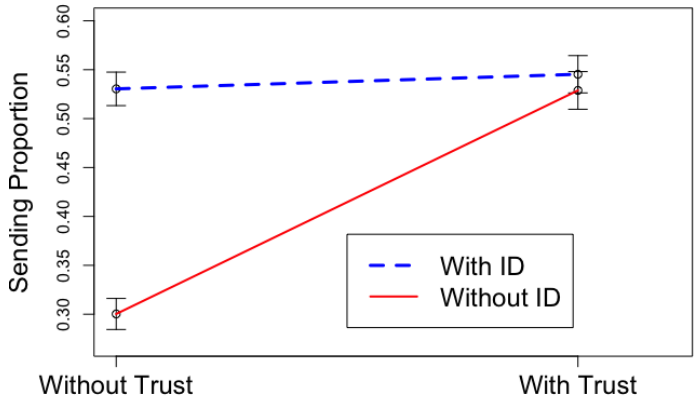


Figure 2.3: Interaction between trust score and ID availability for sender. The bars present standard errors.

sent for each game[6]. The interaction between the availability of trust score and ID on average sending proportion for senders appears in Figure 2.3. We note that showing either trust score or ID improves sending proportion but showing both partner information sources does not change the sent amount relative to one source. Table 2.5 shows the descriptive results by game. The open-jaw pattern suggests the need for paired comparisons between games.

[Johnson and Mislin, 2011] claimed that in large-scale the send proportion of users in trust game follows the normal distribution. We use confidence intervals from paired t-test (yoking by sender ID) in Table 2.6 to document the presence of differences between the Simple Game and any other tested game[7], demonstrating either trust score or ID increases sending amounts with no additive effect. The differences between the other three games (Identity, Score and Combine Games) are not significant, i.e. $p > 0.10$. To rule out any possible difference between sender performance with Show-ID and Show-Trust, we followed up with a paired t-test, yoking the results from the Identity Game and the Score Game for each sender-receiver pair for each trial $t(266) = -0.175$, $p > 0.10$.

Metric	Without Trust		With Trust	
	Without ID (Simple)	With ID (Identity)	Without ID (Score)	With ID (Combine)
Sending proportion by senders	0.30	0.53	0.529	0.545

Table 2.5: Average sending proportion for senders by game

Along with paired t-test, we also tested the difference between games by using post-hoc tests [Crawley, 2015]. Particularly, we used Tukey-HSD test [Kanji, 2006] to verify the difference

[6]We have replicated our findings by using an arcsine transformation $F(1,29) = 16.39$, $p < 0.001$. Because the arcsine transformation is controversial and does not change the pattern of findings, we do not generally report these values.

[7]The negative signs indicate that the sending amount of participants in Simple Game is less than the sending amount of these participants in other games.

Game in Comparison with Simple Game	95% confidence interval	Df
Identity Game	(-0.32, -0.14)	29
Score Game	(-0.35, -0.13)	29
Combine Game	(-0.35, -0.14)	29

Table 2.6: Paired t-based confidence intervals for senders' sending proportion in Simple Game compared to other games

between games for each role. We display the results of Tukey-HSD test in Table 2.7. The Tukey-HSD test confirmed the t-test.

Game	Difference	Lower	Upper	p-value
Identity-Combine	-0.01493333	-0.08006388	0.05019721	-
Score-Combine	-0.01653333	-0.08166388	0.04859721	-
Simple-Combine	-0.24506667	-0.31019721	-0.17993612	***
Score-Identity	-0.00160000	-0.06673055	0.06353055	-
Simple-Identity	-0.23013333	-0.29526388	-0.16500279	***
Simple-Score	-0.22853333	-0.29366388	-0.16340279	***

Table 2.7: Tukey-HSD test for sending proportion of sender in four games.

Using the informal notation, we conclude $IdentityGame \approx ScoreGame \approx CombineGame > SimpleGame$ for sending proportion[8].

2.3.1.2 Cooperative Behavior

Below we address the claim that providing identification or trust score controls cooperative behavior, explaining the above results. We consider the cases of non cooperation where senders send 0, the change in trust scores over time and the dependence of sending behavior on trust score values.

The percentage of times that a sender sends 0 in Simple Game, Identity Game, Score Game and Combine Game are 33.3%, 9.3%, 13.6% and 12.7% respectively. We verified the difference by performing a logistic regression on the frequency of 0 transactions for all rounds with sending participant, Show-Trust and Show-ID as predictors. The logistic regression indicates an interaction between Show-Trust and Show-ID $z = 5.607$, $p < 0.001$. It is more likely that senders send 0 in Simple Game[9]

To examine the potential change in sending behavior over time, we regressed sending behavior on participant ID to remove general participant effects that would contaminate a regression analysis. We then used the resulting residuals as the criterion in a regression with round number as the predictor, reducing the df in the error term due to the prior regression. The only game with a significant round effect was the game with no information (Simple Game), revealing decreasing cooperation over time $F(1,116) = 7.3$, $p < 0.01$. No other game indicated a round effect: Identity Game, $F(1,114) = 0.05$, $p > 0.10$, Score Game $F(1,115) = 0.42$, $p > 0.10$ and

[8]We also replicated these analyses with the non-parametric Kolmogorov-Smirnov (K-S) test for percentage data due to the potential violations of normality, using trial-level data. The K-S test confirmed the findings.

[9]Apart from demonstrating the effect of our manipulations on cooperation, these results preview the reduced and variable degrees of freedom (df) in the analysis of receiver behavior as we removed the 0 transactions.

Combine Game $F(1,116) = 0.008$, $p > 0.10$. Partner information in general eliminates decreasing cooperation over time and end game effects for senders.

	Without Trust		With Trust	
	Without ID (Simple)	With ID (Identity)	Without ID (Score)	With ID (Combine)
Own trust	12.80***	9.31***	7.36***	8.33***
Partner trust	1.65	1.73	5.69***	4.69***
Adjusted R^2	0.85	0.75	0.88	0.89
F(2,27)	86.03	43.57	106.9	117.1

Table 2.8: Trust regression analysis for average sending behavior of senders. The table reports on t values. '*' $p < 0.05$, '**' $p < 0.01$, '***' $p < 0.001$.

Finally, in Table 2.8 we present regression analyses between average sending behavior as the dependent variable with trust values and trust values of sender (participant) and receiver (partner) as predictors. Sender behavior is positively correlated with his own trust value for all games. The trust function predicts sender behavior well. Moreover, partner trust controls sending behavior when it is available. Notably, this is the only analysis suggesting any difference between the availability of partner identity and the trust score, as partner trust score does not predict send behavior in games without a trust score. We conclude that partner trust score availability controls cooperation. We also note the relatively high adjusted R^2 for the Simple Game. We attribute this to range restriction on trust score values that eliminates non-linear influences at higher levels of trust.

2.3.1.3 Summary of Sender Behavior

Senders are less cooperative in the Simple Game than all other games. Decreasing cooperation in the form of round effects only appears in the Simple Game. Good models for sending behavior show predictive effects of own trust in all conditions, and partner trust when trust scores are available. The availability of partner trust score therefore controls sending behavior.

2.3.2 Receiver Behavior

In this section we study the behavior of receiver (trustee) responding to our manipulations. We show that both trust score and ID increase sending generosity with equivalent improvement and no combined effect. To examine cooperation, we analyze the 0 exchange condition[10]. We rule-out round effects and examine the dependence of performance on trust score metrics.

2.3.2.1 Omnibus ANOVA

We performed a basic ANOVA with Subject, Show-Trust and Show-ID as predictors. The ANOVA reveals an interaction, $F(1,29) = 14.36$, $p < 0.001$ as measured by average sending proportion[11]. The interaction between the availability of trust score and ID on average sending

[10] We recall that, the 0-exchange from a receiver means that she received a positive amount from the sender but decided to send back 0.

[11] Similar with the sender case, we performed the same analysis on arcsine transformation, $F(1,29) = 14.74$, $p < 0.001$.

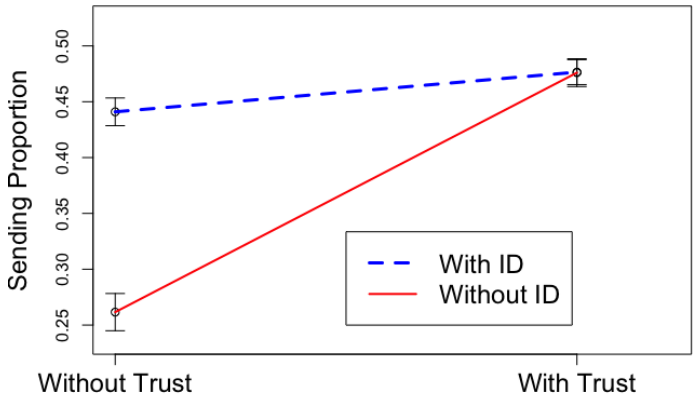


Figure 2.4: Interaction between trust score and ID availability for receiver. The bars present standard errors.

proportion appears in Figure 2.4. We note that showing either trust score or ID improves receiver return proportions, but showing both partner information sources does not change the sent amount relative to one source. The open-jaw pattern suggests the need for paired comparisons between games. Table 2.9 shows the descriptive results by game. As above, and consistent with [Johnson and Mislin, 2011] we assume that the sending proportion of receivers follows the normal distribution in large-scale. We used paired-t based confidence intervals in Table 2.10 to document the absence of differences between the Simple Game and any other tested game[12]. Showing either trust score or ID increases the amount sent back with no additive effect. To rule out any possible difference between receiver performance with Show-ID and Show-Trust, we followed up with a paired t-test yoking the results from the Identity Game and the Score Game for each receiver-sender pair for each trial. The results of the paired t-test, i.e. $t(219) =$ -0.458, $p > 0.10$ confirmed the absence of difference between Identity Game and Score Game.

We verified the difference between games by performing post-hoc Tukey-HSD test. We display the results in Table 2.11. The Tukey-HSD test confirmed the t-test.

We conclude $CombineGame \approx ScoreGame \approx IdentityGame > SimpleGame$ for sending back proportion of receiver behavior.

Metric	Without Trust		With Trust	
	Without ID (Simple)	With ID (Identity)	Without ID (Score)	With ID (Combine)
Sending proportion by receivers	0.262	0.441	0.476	0.477

Table 2.9: Average sending proportion for receivers by game

[12]The negative signs indicate that the sending amount of participants in Simple Game is less than the sending amount of these participants in other games.

Game in Comparison with Simple Game	95% confidence interval	Df
Identity Game	(-0.23, -0.10)	29
Score Game	(-0.25, -0.08)	29
Combine Game	(-0.26, -0.11)	29

Table 2.10: Paired t-test confidence intervals for receivers sending proportion in Simple Game with other games

Game	Difference	Lower	Upper	p-value
Identity-Combine	-0.0355430588	-0.08109565	0.01000953	-
Score-Combine	-0.0004646928	-0.04656710	0.04563771	-
Simple-Combine	-0.2149111883	-0.26433187	-0.16549050	***
Score-Identity	0.0350783661	-0.01065251	0.08080924	-
Simple-Identity	-0.1793681295	-0.22844241	-0.13029385	***
Simple-Score	-0.2144464955	-0.26403156	-0.16486143	***

Table 2.11: Tukey-HSD test for sending proportion of receiver in four games.

2.3.2.2 Cooperative Behavior

Below we address the claim that providing identification or trust score increases cooperative behavior, explaining the above results. We consider the cases of sending 0, the change in trust scores over time and the dependence of receiver behavior on trust score values

The percentage of times that a receiver sends 0 when she received a positive amount from sender in Simple Game, Identity Game, Score Game and Combine Game are 36.8%, 8.5%, 8.3% and 4.5% respectively. We performed a logistic regression on the frequency of 0 transactions for all trials with sending participant, Show-Trust and Show-ID as predictors. The logistic regression indicates an interaction between Show-Trust and Show-ID $z = 3.68$, $p < 0.01$. Receivers are more likely to return 0 in the Simple Game.

To examine the potential change in receiver behavior over round, we regressed receiver behavior on participant id to remove general participant effects that would contaminate a regression analysis. We then used the resulting residuals as the criterion in a regression with round number as the predictor, reducing the df in the error term due to the prior regression. Round is not significant for any game: Simple Game $F(1,100) = 0.052$, $p > 0.10$, Identity Game, $F(1,114) = 1.44$, $p > 0.10$, Score Game $F(1,108) = 0.019$, $p > 0.10$ and Combine Game $F(1,110) = 0.027$, $p > 0.10$. Participant information conditions therefore have no effect on the prevention of end-game effects.

Finally, in Table 2.12 we present regression analyses between average sending behavior as the criterion with sender trust values, participant trust values and amount received from the sender as predictors. Receiver behavior is positively correlated with his own trust value for all games. This confirms our ability to predict receiver cooperation (i.e., receiver trustworthiness) from past trust values. However, receiver behavior is only related to partner trust in the combined game. Moreover, model fits are not as good for receivers as they are for senders[13].

[13]We have explored models that include interactions between amount received and trust values. These often improve the relatively smaller adjusted R^2 we obtain for receiver behavior. Such models suggest the need for different trust functions for sender and receiver, to accommodate the asymmetry in their relationship.

	Without Trust		With Trust	
	Without ID (Simple)	With ID (Identity)	Without ID (Score)	With ID (Combine)
Own trust	6.003**	8.936***	4.617***	3.927***
Partner trust	0.687	0.978	0.237	-2.158*
Partner sending amount	-2.214*	-1.849	-1.469	0.587
Adjusted R^2	0.565	0.746	0.415	0.494
F(3,26)	13.53	29.36	7.854	10.44

Table 2.12: Trust regression analysis for average sending behavior of receivers. '*' $p < 0.05$, '**' $p < 0.01$, '***' $p < 0.001$.

2.3.2.3 Summary of Sender Behavior

Receivers are less cooperative in the Simple Game than all other games. There is no evidence of round effects in any game. Fair models for sending behavior show predictive effects of own trust in all conditions confirming our trustworthiness predictions. However, partner trust is only predictive in the combined game.

2.4 Experimental Design Issues

In this section we investigate the properties of our experiment, comparing our results with other trust game experiments, evaluating the accuracy of our trust function, and addressing repeated measures concerns such as the nesting of participants in groups.

2.4.1 Comparison with other trust game data sets

Departures from the standard trust game require us to establish that our findings are not due to such idiosyncrasies rather than the manipulations we have examined.

We compared the average sending proportions of participants in our Simple Game (30 data points) with two external datasets from [Dubois et al., 2012] with 36 data points and [Bravo et al., 2012] with 108 data points. Table 2.13 shows Welch two-sample t-test values comparing our results in the simple game to their results, assuming unequal variances. None of the comparisons are statistically significant. The observed behavior in the simple game in our experimental design is consistent with other experiments. We visualized the findings in Figure 2.5.

	Dubois (2012)	Bravo (2012)
Sender	t(61.6) = -1.33	t(45.3) = -0.991
Receiver	t(55.9) = 1.69	t(45.6) = -0.598

Table 2.13: Welch two-sample t-values between our Simple Game average send proportion data with two external datasets.

2.4.2 Trust function analysis

In the previous sections, we demonstrated that showing the trust score improves cooperation, but how good is the trust function? If merely showing a score can improve the behavior of

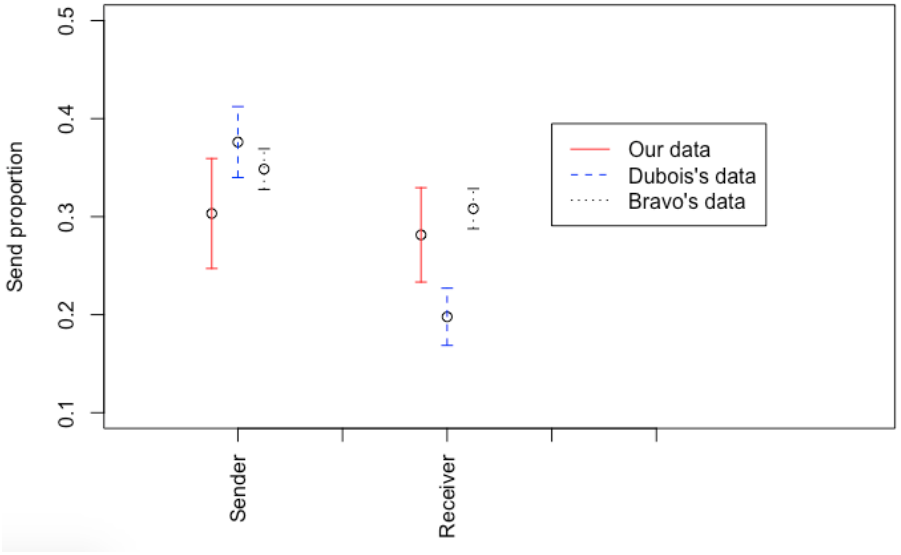


Figure 2.5: Visualization of the average values and standard errors of users' sending proportions in three datasets.

the participant, perhaps any random number would suffice. We provide two forms of support for the quality of the trust function: prediction of participant behavior in our experiment and prediction of participant behavior in two external datasets.

2.4.2.1 Predicting behavior in our experiment.

The trust score models participant behavior, even when, as in Simple and Identity Games, the trust score is not made available to participants. Thus participant behavior should correlate with their own trust scores. In the games with available trust scores (Score and Combine Games), participant behavior should appear to react to partner trust values. The R^2 values in Tables 2.8 and 2.12 provide some evidence of prediction accuracy, although we noted less satisfactory models for receivers, and less evidence for the relevance of partner trust values in receiver behavior. Here we rule out interactions between trust values themselves as better predictors of behavior. We also examine correlations between behavior and trust scores separately for rounds 4 and 5 when trust scores have sufficient data to stabilize.

Regressions of sender behavior, i.e. average sending proportion, on the interaction of sender and receiver trust values in the presence of both predictors as main effects provide no evidence of interaction effects in any game: Score Game $t(26) = 1.079$, $p > 0.1$, Combine Game $t(26) = 0.022$, $p > 0.1$, Simple Game $t(26) = -0.352$, $p > 0.1$ nor Identity Game $t(26) = 0.725$, $p > 0.1$.

Regressions of receiver behavior, i.e., average return proportion, on the interaction of sender and receiver trust values in the presence of both predictors as main effects provide no evidence of interaction effects in any game: Score Game $t(26) = -0.122$, $p > 0.1$, Combine Game $t(26) = -0.776$, $p > 0.1$, Simple Game $t(26) = 0.706$, $p > 0.1$ nor Combine Game $t(26) = 0.080$, $p > 0.1$. Adding interactions between trust predictors does not improve our models.

To further examine the predictive power of the trust function, we performed separate multiple regression analyses for each game, for rounds 4 and 5 when trust scores have accrued sufficient data. The dependent variable is the sending proportion of the participants to their partners. Table 2.14 provides the results of a regression of the senders sending proportion on a model with

her trust value and the trust value of her partner for both rounds. In all cases, the sender's trust value predicts sending behavior. Moreover, the partner's trust value also predicts sending behavior in the presence of ID or trust score information, confirming sender attention to these sources. Adjusted R^2 values range from 0.26 to 0.70, with lower values resulting from the game with no information.

Table 2.15 provides comparable information for receiver behavior, answering the question of how well we can predict whether a participant is trustworthy. These regression models included own trust value, partner trust value and the amount just received (i.e., three times the amount sent). While receivers never were aware of their own trust values, our trust function is a good predictor of receiver behavior *when trust score is not provided.* This does support our claim that the trust function is a good predictor of trustworthiness. However, the mere presence of trust scores in the trust score conditions dampens its predictive capability. Partner trust value is rarely predictive. Receivers did not rely on this systematically. Adjusted R^2values range from 0.08 to 0.45 with higher values in the conditions where trust score is *not* provided.

	Without Trust		With Trust	
	Without ID (Simple)	With ID (Identity)	Without ID (Score)	With ID (Combine)
Round 4	df = 72	df = 72	df = 72	df = 72
Own trust value	6.46***	5.80***	3.89***	7.28***
Partner's trust value	0.67	3.24**	6.98***	4.41***
Adj. R^2	0.36***	0.40***	0.66***	0.70***
Round 5	df = 72	df = 72	df = 72	df = 72
Own trust value	4.87***	7.13***	3.19**	7.11***
Partner's trust value	1.16	4.54***	7.38***	3.52***
Adj. R^2	0.26***	0.55***	0.67***	0.70***

Table 2.14: Trust regression analysis on senders' sending proportion with t-values for individual slope tests. '*' $p < 0.05$, '**' $p < 0.01$, '***' $p < 0.001$.

2.4.3 Post-hoc Reputation Analysis

In this section, we present a post-hoc analysis to compare the predictive power of future behavior of participants in the trust games we designed for our experiment between trust and reputation scores.

In our analyses presented in Tables 2.16 and 2.17 we substituted reputation predictors for trust predictors, using average sending proportion as the criterion. These models differ from those in Tables 2.8 and 2.12 by the absence of own-score predictors. These reduced models were necessary because of the close relationship between average reputation and average sending amount. However, the absence of own-values does inflate the error term. As in Table 2.8, in Table 2.16 partner values predict sender behavior when trust values are shown. As measured by Adjusted R^2, the resulting models of sender behavior with trust predictors are better than models with reputation predictors. Regarding receiver behavior, in Table 2.12 partner trust is only significant in the combined game. In Table 2.17 partner reputation predicts receiver behavior for the ID game, no doubt assisted by the significant effect of partner sending amount. We note that in those cases with significant partner effects, the direction is negative regarding to

	Without Trust		With Trust	
	Without ID (Simple)	With ID (Identity)	Without ID (Score)	With ID (Combine)
Round 4	df = 42	df = 62	df = 60	df = 60
Own trust value	3.41**	7.21***	1.98	1.76
Partner's trust value	0.02	1.40	1.63	0.50
Amount received	-0.53	-1.62	-2.37*	0.33
Adj. R^2	0.18*	0.45***	0.08	0.10*
Round 5	df = 39	df = 61	df = 61	df = 60
Own trust value	4.21***	3.56***	3.06**	1.09
Partner's trust value	0.14	2.10*	0.74	1.53
Amount received	-2.19*	0.06	-1.75	-0.16
Adj. R^2	0.30***	0.29***	0.13*	0.09*

Table 2.15: Trust regression analysis on receivers' sending proportion with t-values for individual slope tests. '*' $p < 0.05$, '**' $p < 0.01$, '***' $p < 0.001$.

the amount received. Model fits are poor. Adjusted R^2 are however better for trust predictors than the reputation predictors for the games where trust information was present.

	Without Trust		With Trust	
	Without ID (Simple)	With ID (Identity)	Without ID (Score)	With ID (Combine)
Trust predictors				
Partner trust	1.09	0.33	7.42***	6.92***
Adjusted R^2	0.007	-0.03	0.65	0.62
F(1,28)	1.202	0.11	55.07***	47.86***
Reputation predictors				
Partner reputation	0.69	-1.14	4.55***	3.78***
Adjusted R^2	-0.01	0.01	0.40	0.31
F(1,28)	0.48	1.3	20.72***	14.31***

Table 2.16: Trust and reputation analysis for average sending proportion of senders. The table reports on t values. '*' $p < .05$, '**' $p < .01$, '***' $p < .001$.

2.4.4 Group Effects

While data on the trust game are typically collected in groups, concern for group effects has received little attention in trust game analyses. Moreover, in our experiment, group is confounded with treatment order. In order to consider group effects, we conducted a three factors split-plot ANOVA with group as a between subjects effect and Show-ID and Show-Trust as within subjects effects [Keppel, 1991]. Moreover, if group is regarded as a random (sampled) factor, then the independent variables are properly tested against the interaction of group with the independent variables.

	Without Trust		With Trust	
	Without ID (Simple)	With ID (Identity)	Without ID (Score)	With ID (Combine)
Trust predictors				
Partner trust	-0.71	-0.41	-0.26	-2.73*
Partner sending amount	-0.22	1.35	0.85	3.20*
Adjusted R^2	0.00	0.00	-0.02	0.22
F(2,27)	0.99	1.05	0.64	5.18*
Reputation predictors				
Partner reputation	-1.70	-2.72*	-0.15	-1.40
Partner sending amount	0.23	2.33*	0.45	2.07*
Adjusted R^2	0.08	0.21	-0.02	0.08
F(2,27)	2.26	4.93	0.62	2.21

Table 2.17: Trust and reputation analysis for average sending proportion of receivers. The table reports on t values. '*'$p < .05$, '**' $p < .01$, '***' $p < .001$.

Our sole concern here therefore is the robustness of manipulation effects in a very conservative, low power test owing to the reduced df in the error term. We tested our effects considering group as a random factor, and interactions with group as an error term. Our analysis of sending behavior, as measured by relative sending proportion, withstands even this less powerful test. The omnibus test for the interaction of ID and Trust is $F(1,4) = 8.86$, $p < 0.05$. Moreover, none of the Group by Treatment interactions are significant: with Show-Trust $F(4,25) = 2.610$, $p > 0.05$, with Show-ID $F(4,25) = 1.253$, $p > 0.05$, or the interaction $F(4,25) = 2.698$, $p > 0.05$[14] Regarding receiver behavior, as measured by relative returned proportion, the omnibus interaction contrast just misses significance $F(1,4) = 6.966$, $p < 0.1$. These findings are best captured as two main effects: for Show-Trust $F(1,4) = 74.44$, $p < 0.001$ ($M =$ and for Show-ID $F(1,4) = 35.862$, $p < 0.01$). As above, none of the Group by treatment interactions are significant: with Show-Trust $F(4,25) = 0.153$, $p > 0.75$, with Show-ID $F(4,25) = 0.553$, $p > 0.75$, or the interaction $F(4,25) = 2.484$, $p > 0.05$.

These analyses limit concern for group effects in general, and the game order differences confounded with group in particular[15].

.

2.5 Discussion

Below we consider our findings with respect to our research questions, system design implications and limitations.

[14]Precautionary adjustments for sphericity are not required because all repeated factors have one degree of freedom [Winer et al., 1971, page 306].

[15]We conducted similar analyses for sending behavior by trial for the first eight trials, which only revealed a single significant case of Group by Treatment interactions in 24 tests.

2.5.1 Summary

In the trust game senders and receivers have two different roles and potentially behave differently with respect to the provision of partner information. We analyzed some concerns and findings distinguishing between two roles.

Result 1 *Does showing partner trust score or ID change user cooperative behavior?*

We provided several forms of evidence regarding the influence of these interventions on cooperation. These include overall increases in the proportion returned and reductions in the frequency of 0 unit returns for both senders and receivers. Only the simple game differs from the alternatives, in paired-t tests of sending behavior and in the persistence of end-game effects for senders. Otherwise, we eliminated end game effects. Large-n, yoked dependent t-tests by round failed to reveal any difference in behavior between the availability of names and the availability of trust scores.

Result 2 *Does the trust calculation predict participants' future behavior ?*

Our models are generally more successful for predicting sender (trustor) behavior, although some findings predict receiver (trustee) behavior.

With respect to senders, we provide excellent predictive models for average behavior. These average models always depend positively on own trust values, and on partner trust values when trust values are available. Sender behavior is also well modeled at the round level, always depending upon own trust values and on partner trust values for all games except the simple game. Senders are attending to the specific values shown for partners, as predictions based on reputation are not as good as predictions based on the trust values displayed. We note that the effect is not to encourage blind cooperation, but rather cooperation in response to the available information. Low partner trust scores elicit low sending amounts.

With respect to receivers, models of average return proportions behavior do depend on own-trust. This supports a claim for some ability to predict trustworthiness. Models at the round level are best when the trust score is *not* available. This unexpected result is possibly due to strategic differences in receiver behavior. Models are quite poor when own-values are removed in order to compare with reputation predictions. While receiver models did include an additional factor (partner sending amount), our general impression is that the models of receiver behavior are more complex than models of sender behavior and not yet accommodated by the trust function used. Moreover, unlike the sender, duplicitous receiver behavior is not punished until the subsequent round. These considerations suggest that the trust function should differ for sender and receiver.

We have not identified the source of leverage on the success of the trust function for senders. Relative to an average reputation calculation, we have noted three different influences: the specification of partners, the management of change over time and the treatment of variability, particularly punishment in response to non-cooperative behavior. These influences cast the trust function as a *psychometric* issue, concerning the psychological factors that influence the response to experience. Limitations in the receiver model highlight this claim, where the role of amount received may interact with the partner trust values in ways that we have not yet captured.

Certainly, other dimensions merit investigation. The relationship between age, gender and behavior in trust game is not established in the literature: several studies claimed no relation [Cesarini et al., 2008; R. Slonim and Garbarino, 2008; R. Slonim and Guillen, 2010]. Other research claimed that men trusted more than women in sender role, and less in receiver role [Buchan et al., 2008] but other studies refute this finding[Haselhuhn et al., 2015].

The trust function used considers only the sending proportion as a parameter, but not for instance the amount sent by the partner. This trust model fits well for a sender that initiates

the interaction by sending an initial amount. But the trustworthiness value associated to a receiver should depend not only on the return proportion but also on the amount received. We might consider associating a higher trustworthiness to a receiver that received 6 and returned 0.5 than to someone that received 30, but returned the same proportion. The receiver that received 30 received the maximum possible amount but did not reciprocate the granted trust. These suggestions further reinforce the need to consider the measurement of trust from a psychometric perspective, capturing the relationship between physical quantities and behavioral response.

2.5.2 System Design Implications

We have demonstrated that the presence of partner information benefits cooperative behavior. The burden of recalling past experience with participants is just one justification for the use of trust values as a source of this information [J. Tang, X. Hu, et al., 2013].

Compared with reputation scores, trust scores have several advantages. Reputation scores are globally computed values that are stored on a central server that is vulnerable to attack [Hoffman et al., 2009; Sun and Ku, 2014]. Trust scores are suitable for *distributed* architectures and do not require a central server. Trust scores are computed in a distributed way for each user: each member of the network locally computes trust levels of her partners. Moreover, trust scores emphasize *personal* experience and value. For instance, in reputation systems, if ten thousand participants rated a seller, the next participant does not have a high motivation to provide a rating because it will not change the average rating score of this seller. However, in trust-based systems, her impression has a great influence because the trust value is calculated for her only based on her experience.

On the other hand, as our experiments suggested, the trust score has a similar effect on cooperative behavior relative to ID. Therefore, the trust scores may complement current systems that employ ID to identify users, helping users define the trustworthiness of their connections. While it is possible for participants to change their ID in on line systems, they cannot not change the trust level other participants assigned to them. If a trust score is available, participants do not need to remember individuals by name, nor do they need to assess previous experience with imprecise mental calculations. Instead, they can make decisions based on their partner's current trust score.

Such a system greatly facilitates engagement with large scale collaborative networks. Our proposed solution for computing partner trust scores scales well with the number of partners. For each user u_i, where $1 \leq i \leq n$ and n is the total number of partners, the system stores m_i trust values t_{ij}, with $1 \leq j \leq m_i$, associated with the m_i partners with whom he is interacting. Each time a participant u_i interacts with another parnter u_j, the trust score corresponding to that interaction is aggregated to the old trust value t_{ij}. The new aggregated value becomes the new value of t_{ij}. The time complexity of the computation of the trust score from an interaction is $O(1)$, i.e. constant. The space complexity for a participant to keep track of the trust scores of the other participants is linear with the number of participants with whom he interacts.

2.5.3 Limitations

Limitations span issues of experimental design and issues of generalizability.

Power is a possible consideration in the failure to identify a difference between the three experimental conditions. We addressed this with large-n analyses at the round level. Moreover, our sample size is consistent with [Dubois et al., 2012] who organized a team of 36 participants. Small group sizes ($4-6$ people per group) are commonly observed in the experimental trust game

[Bohnet and Zeckhauser, 2004; Gary E. Bolton et al., 2005; Camera and Casari, 2009], mostly because of practical difficulty in recruiting and organizing participants. The total number of participants is usually limited also. For instance, [Lunawat, 2013] organized experiments with 16 and 22 participants. Finally, we note that inflating sample size to force a difference is likely to result in a small effect size.

Few studies criticized trust game for the lack of context [Riegelsberger, Martina Angela Sasse, et al., 2005]. Our view, consistent with the proponents of situated cognition, is that there is no such thing as an absence of context. Games requiring limited background knowledge control for individual differences in expertise and provide statistical power. We view the use of a standard paradigm as crucial to our exploratory studies. Behavior in this paradigm is well-documented, with known pitfalls such as end-game effects, and known standards for cooperation. Because we obtained results in the simple game that are consistent with the literature, we can attribute our findings to our interventions, rather than idiosyncrasies of an unknown paradigm.

Regarding generalizability, significant effort remains in developing trust functions for other domains. Whether the issue is commercial trade, sharing information or granting modification access, the interaction requires a quantitative foundation. Our claim is not that the specific function we used is suitable for every domain, but rather that the dimensions we have identified (partner specificity, the representation of cumulative experience, and the treatment of variability) are candidates for inclusion. As we discuss in Chapter 3, our trust function can be applied into Wikipedia. We claim that the proposed trust model is able to generalized.

2.6 Extension of Experimental Results

The experimental results in trust game suggests the positive effects of showing trust score to users in encouraging them to collaborate more. However, it is not clear that the same effect will occur if we introduce trust score in real-world systems like Wikipedia.

There is no certain answer until we can validate the influence of trust score in the real-world systems, but as we discussed above it is very costly and almost impossible to deploy and test in real scenarios. However, based on the long history of experimental behavior study [Pruitt and Kimmel, 1977; Wilde, 1981; Kendall et al., 2007], studies have suggested that the experimental results in studying human behavior can be applied into real-world scenarios [Falk and Heckman, 2009] if the appropriate adjustments are provided [J. List, S. Levitt, et al., 2007]. In other words, the experimental results of lab-control experiments provide a general guideline in principle about human behavior but not a details instruction of how to implement them in real-world scenarios.

On the other hand, the lab-control experiments are used because their suggestions, if any, are independent from the context, hence for each real-world scenarios the engineers can find a different way to deploy the suggestions. For instance, if we validated the influence of trust score on user behavior in Wikipedia, the results might not be extendable to Google Docs. The first reason is that the trust score will be very likely to be deployed along with other existing mechanisms such as nick-name, avatar, etc. and these existing mechanisms are different between systems. The second reason is that the interaction of users in Wikipedia will be much more complicated than in trust game to analyze the causality between trust score and the changes in user behavior. [Charness and Kuhn, 2011] discussed in details about gaming experiments and claimed that the experiments are important tools to study human behavior, and the results from the experiments can be used externally. [Baran et al., 2010] particularly address the question of inferring the social preferences from lab data. The authors found the consistent between behavior of same participants in trust game and real-world: who sends more in trust game

tends to donate more in real-world.

Several observations in lab-control gaming experiments in general and in trust game in particular have been confirmed in real-world scenarios. [Benz and Meier, 2008] in Zurich and [Baran et al., 2010] in Chicago particularly addressed the question of inferring the social preferences from lab data. The authors found the consistent between behavior of same participants in trust game and real-world: who sends more in trust game tends to donate more in real-world. [Karlan, 2005] studied the difference of behavior in trust game and in real-life of people in Peru while [Johansson-Stenman et al., 2013] made a similar research work in Bangladesh. Both research studies confirmed that the results from trust game experiments are consistent with the results from field studies [S. D. Levitt and J. A. List, 2009]. [Yao and Darwen, 1999; Gary E Bolton et al., 2002] observed the effects of reputation score on user behavior in repeated trust game and the effect has been confirmed in eBay [Resnick, Zeckhauser, et al., 2006]. Similarly, the influence of partner's avatar on decision of users are both observed in real-world systems [Pentina and Taylor, 2010] and in gaming experiences [Wilson and Eckel, 2006; Bente, Rüggenberg, et al., 2008]. [J. Zheng et al., 2001] studied the effect of *chat* on improving trust between users in prisoner-dilemma while [A Ben-Ner et al., 2009] studied the effect of chat in repeated trust games, and the effect of chat has been confirmed in collaborative software development environment [Hupfer et al., 2004].

We showed in repeated trust game, showing trust score to users will encourage the collaboration between users. We showed that users follow trust score. We analyzed and argued that trust score can overcome some limitations of popular techniques such as nick-name, avatar and reputation score. We conclude that trust score should be deployed in real-world collaboration systems. In the next chapter, we discuss in details the trust model, i.e. how do we calculate the trust score of users in collaboration.

Chapter 3

Measuring Trust: Case Studies in Repeated Trust Game and Wikipedia

> The best material model of a cat is another, or preferably the same, cat.
>
> — A. Rosenblueth & N. Wiener, Philosophy of Science (1945)

Contents

In the previous chapter, we answered the first research question: "Should we introduce trust score to users?". We showed that a computational trust model can be deployed to assist users in assessing the trustworthiness of partners. However, we did not discuss how do we calculate trust scores of partners to display to users. In this chapter, we discuss the next question: "How do we calculate trust score of users in a collaborative system?"

Studies suggested that different contexts require different trust models [Huynh, 2009]. Several collaborative systems exist already as of this writing so we can not cover all of them in this thesis. We focus on two contexts that are trust game and Wikipedia.

Trust game is a lab-control collaborative environment. Studying trust method design in trust game could give us some more insight for designing trust methods in real-world settings.

As we discussed in Section 1.3, the findings about human behavior in trust game can be applied in other real-world systems.

On the other hand, Wikipedia is a result of a tremendously collaboration effort from millions of users. As we presented in Section 1.3, studies emphasized the importance of Wikipedia to Internet users. Designing a trust method in Wikipedia could help to speed up the review process therefore new information can be published faster on Wikipedia.

3.1 Trust Calculation in Repeated Trust Game

In this section, we present our trust calculation method which has been deployed to test the influence of trust score on user behavior in the previous chapter. We used the same datasets described in the previous chapter to validate the trust method.

3.1.1 Trust Calculation

Trust score is defined as a metric to measure the goodness of user behavior in the past. A trust model is considered as a good trust model if we can use the score calculated by this model to predict future behavior of users. On the other hand, a trust model is considered as a failed model if it assigns a high trust score to a user but this user deviate in the future, because in this scenario the trust model failed to give a warning message to other users on a malicious user.

In trust games, trustworthiness of a user depends on the amount sent to her partners [Dubois et al., 2012; Glaeser et al., 2000]. A higher sending amount should lead to higher trustworthiness, but the relationship between these two variables is not necessary linear. As we discussed in Section 2.1.1.3, a user might try to behave well in the beginning and then suddenly deviate. Our trust calculation will take into account this strategy that we call *fluctuate strategy*.

The trust value calculated by the trust model needs to satisfy the following requirements:

1. The trust value is higher if the sending amount is higher.
2. The trust value can distinguish between different types of users.
3. The trust value considers user behavior over time.
4. The trust value encourages a stable behavior rather than a fluctuating one.
5. The trust value is robust against attacks.

The final trust score of a user is combined from several scores as follows.

3.1.1.1 Current trust

In repeated trust game, for each round, two users who are *sender* and *receiver* interact by sending non-negative amounts to each other. The maximum amount the sender can send is 10, and the maximum amount the receiver can send is the amount she received from the sender (i.e. three times of what the sender sent). For both roles, we normalize the $send\_proportion_t$ as the sending proportion of a user at round t, with $t \geq 1$:

$$send\_proportion_t = \frac{sending\_amount_t}{maximum\_sending\_amount_t} \tag{3.1}$$

In case of receiver, if the sender sent first the amount of 0, the receiver has no other option but sending back 0 also. We eliminate this zero-transaction for receiver, because of two reasons: (i) the behavior of sending back 0 is not informative, i.e. it does not tell us any new information about the receiver, and (ii) the *send_proportion* which is 0/0 is not calculable. In this case, we keep the previous trust score of the receiver to the next round. We note that the zero-transaction elimination is applied only for receiver. In case the *send_proportion* value is calculable, it is easy to see that $\forall t, 0 \leq send\_proportion_t \leq 1$.

The trust score calculated for a single interaction between users is called *current_trust*. $current\_trust_t$ is defined as a function of $send\_proportion_t$. We define $current\_trust_t \in [0, 1]$. This function should satisfy the following properties (for convenience, we use the notation $f(x), f : [0, 1] \rightarrow [0, 1]$ for the function of $current\_trust_t$, with x being $send\_proportion_t$):

- $f(x)$ is continuous in $[0, 1]$.
- $f(0) = 0$, i.e. *current_trust* is 0 if the user sends nothing.
- $f(1) = 1$, i.e. *current_trust* is 1 if the user sends the maximum possible amount.
- $f'(x) > 0$ with $x \in [0, 1]$, i.e. $f(x) > f(y)$ iff $x > y$ for $x, y \in [0, 1]$ meaning that the value of *current_trust* is strictly increasing when *send_proportion* increases from 0 to 1. $f'(x)$ denotes the derivative of function $f(x)$.
- $f''(x) \leq 0$ with $x \in [0, 1]$ meaning that the function is concave, i.e. the closer to 1 the value of *current_trust* is, the harder is to increment it.
- $f'(x^-) = f'(x^+), \forall x \in [0, 1]$, meaning that the function is smooth, i.e. there is no reason that at some point the current trust increases sharply less than previously.

We propose the following function that satisfies the above mentioned conditions:

$$current\_trust_t = \log(send\_proportion_t \times (e - 1) + 1) \tag{3.2}$$

where $current\_trust_t$ is the *current_trust* function at round t and $send\_proportion_t$ is the value of *send_proportion* at round t.

Explanation about the selection of the formula 3.2 will be provided in following sections.

3.1.1.2 Aggregate Trust

$current\_trust_t$ uniquely computes the value of trust based on a single current interaction t. However, we also take into consideration the previous interactions between two users. The calculation of trust for repeated interactions is inspired by the trust model SecuredTrust [Das and Islam, 2012]. The main shortcoming of SecuredTrust is that the metric assumes the existence of $current\_trust_t$ value. However, as shown in the previous section, computing *current_trust* is not a trivial task as it has to satisfy several requirements. Furthermore, SecuredTrust was mainly designed for peer-to-peer network systems where the computation of the trust in a peer node relies on information provided by the neighbours in the network. In this way, the trust value in one peer is in fact the reputation of that peer computed as an aggregation of the neighbor trust values on that peer. Nevertheless, in collaborative environments different users have different experiences with a certain user and therefore their trust values on that user are different.

On the other hand, SecuredTrust uses a constant value α as forgetting factor. If this property can be valid in the peer-to-peer networks, it does not hold for human users. Based on

psychological peak-end rule [Fredrickson and Kahneman, 1993] we present a dynamic α. The peak-end rule claims that, in a series of experiences, humans remember the extreme and the last experience, and tend to forget the other ones.

We calculate *aggregate_trust* as follows:

$$\delta_t = |current\_trust_t - current\_trust_{t-1}| \tag{3.3}$$

$$\beta_t = c \times \delta_t + (1 - c) \times \beta_{t-1} \tag{3.4}$$

$$\alpha_t = threshold + \frac{c \times \delta_t}{1 + \beta_t} \tag{3.5}$$

$$\begin{aligned} aggregate\_trust_t = {} & \alpha_t \times current\_trust_t \\ & + (1 - \alpha_t) \times aggregate\_trust_{t-1} \end{aligned} \tag{3.6}$$

As we describe in Section 3.1.1.4, $current\_trust_0 = 0$. We also present other constant values used for trust calculation in this section.

The δ_t is the measurement of change of current trust values by two sequential interactions $t-1$ and t between two users. We calculate δ_t to see how much a person changes her behavior with a partner since their last interaction. It is easy to prove that, α_t is bigger if δ_t is bigger, and vice versa. It means that, if the trust of the current interaction is much different from accumulated trust of all previous interactions, the current interaction will play a more important role in the final trust value.

3.1.1.3 Dealing with fluctuating behavior

Some users may collaborate in the beginning and then suddenly stop collaborating. We add a $change\_rate_t$ variable into our model to punish this kind of activity.

First, we calculate the $trend\_factor_t$ at round t representing the recent trend of user behavior, with higher value meaning that users improved lately their behavior:

$$trend\_factor_t = \begin{cases} trend\_factor_{t-1} + \phi \\ \quad \text{if } current\_trust_t - aggregate\_trust_t > \epsilon \\ trend\_factor_{t-1} - \phi \\ \quad \text{if } aggregate\_trust_t - current\_trust_t > \epsilon \\ trend\_factor_{t-1} \quad \text{otherwise} \end{cases} \tag{3.7}$$

$$adj\_atf_t = \begin{cases} \frac{atf_t}{2} & \text{if } atf_t > \text{MAX\_ATF} \\ atf_t & \text{otherwise} \end{cases} \tag{3.8}$$

$$atf_t = \begin{cases} adj\_atf_{t-1} + \frac{(current\_trust_t - aggregate\_trust_t)}{2} \\ \quad \text{if } current\_trust_t - aggregate\_trust_t > \phi \\ adj\_atf_{t-1} + (aggregate\_trust_t - current\_trust_t) \\ \quad \text{if } aggregate\_trust_t - current\_trust_t > \phi \\ adj\_atf_{t-1} \quad \text{otherwise} \end{cases} \tag{3.9}$$

$$change\_rate_t = \begin{cases} 0 \text{ if } atf_t > \text{MAX\_ATF} \\ \cos\left(\frac{\pi}{2} \times \frac{atf_t}{\text{MAX\_ATF}}\right) \text{ otherwise} \end{cases} \tag{3.10}$$

We present the accumulated trust fluctuation (*atf*) function in the formula 3.9. We aim to punish both kinds of *fluctuation-based cheating behaviors*. We consider both scenarios when the latest sending amount is suddenly higher or lower than usual behavior as cheating behaviors. However, it is arguable that the latter behavior is more dangerous than the former one. Therefore, the punishment in the latter case will be stronger.

The accumulated trust fluctuation is a non-decreasing function. The increase depends on the change over time of user's behavior. If the behavior is stable or changes within the allowed range (defined by the constant ϕ), atf_t will not change.

When atf_t reaches the threshold value MAX_ATF, it means that accumulated change in user behavior over time reaches the level of betrayal and therefore $change\_rate_t$ drops to 0. Otherwise, as shown by Equation 3.10, $change\_rate_t$ decreases if atf_t increases.

The cosine function is used in formula 3.10 because the cos function has a low degradation rate in the initial stage, and a high degradation rate in the case of repeated fluctuating behavior[Das and Islam, 2012]. It means that, if a user starts adopting a fluctuating behavior the punishment is low, but it increases fast while fluctuating behavior persists.

Finally, we calculate the trust value after round t:

$$\begin{aligned} trust\_value_t &= expect\_trust_t \times change\_rate_t \\ &\text{where,} \\ expect\_trust_t &= trend\_factor_t \times current\_trust_t \\ &\quad + (1 - trend\_factor_t) \times aggregate\_trust_t \end{aligned} \tag{3.11}$$

We update the trust value on each round, after both players made their decisions.

3.1.1.4 Parameters initial values

We display the values of the parameters used for the trust metric computation in Table 3.1. The left side of the table contains the initial values of the corresponding parameters, while the right side of the table contains the constant values of the corresponding parameters. We explain the choice of these initial values in Section 3.1.2.1.

Table 3.1: Parameter Initial values.

α_0	0.	ϵ	0.3
β_0	0.	ϕ	0.1
atf_0	0.	MAX_ATF	2.
$expect\_trust_0$	0.	*threshold*	0.25
$trend\_factor_0$	0.	c	0.9
$current\_trust_0$	0.		
$aggregate\_trust_0$	0.		
$change\_rate_0$	0.		
$trust\_value_0$	0.5		

3.1.2 Trust Model Evaluation

In this section, we evaluate the performance of our trust model according to the following three aspects:

1. *Performance in simulated data.* We evaluate our trust model with simulated data. More specifically, we analyze whether our trust model can distinguish between user types and cope with a fluctuating strategy.

2. *Consistency with human opinions.* We study how we can validate our trust model with real user data. More specifically, given the same data set, we analyze whether our trust metric provides the same ratings of user behavior as the ones manually assigned by humans.

3. *Performance in real data.* We study whether our trust model can predict users future behavior. In other words, we analyze whether the trust score assigned by the trust model to a user reflects her future behavior.

As discussed in Section 1.4.2.2, in general we evaluate a trust model by analyzing whether we can use the trust score computed by this model to predict the future behavior of the partners. However, due to the advantage of trust game as we described in Section 2.1.2, we can study and analyze the consistency of our trust model with human opinion from previous studies.

3.1.2.1 Evaluation with simulated data

Our trust model should punish fluctuating user behaviors. Moreover, it should detect user behavior patterns, i.e. it should be able to distinguish different types of user profiles: low, medium and high which correspond to a user who send at high, medium or low amount in all of their rounds. In this section, we verify that our trust model satisfies these criteria.

Fluctuating user behaviors We define three types of user profiles according to the values of *send_proportion*: low, medium and high. Similar to [Buntain and Golbeck, 2015], we define that a user with a low profile sends in average 20% of the maximum possible amount, while for a medium profile user the *send_proportion* is 50% and for a high profile user it is 80%. We also define a *fluctuate profile* user who first tries to behave well and then deviates.

By means of simulations[16] for the above user profiles, we compare the behavior of our trust model with the average trust model which calculates the trust score of a user on a partner as an average of previous sending amount from this user to this partner [Anderhub et al., 2002; Avner Ben-Ner and Putterman, 2009; Cochard et al., 2004; Dubois et al., 2012; Engle-Warnick and Robert L. Slonim, 2006; Glaeser et al., 2000; Johnson and Mislin, 2011]. We display the trust scores calculated by our trust model in Figure 3.1. We display the trust scores calculated by the average model in Figure 3.2.

We can see that, our trust model can cope and punish the *fluctuating* behavior very well, as it reduces the trust score of *fluctuating* user to the same as of a *low profile* user. On the other side, the simple average metric cannot distinguish between *fluctuating* and *high profile* users.

[16] All the data and code including simulation code and analysis code are available at [Q.-V. Dang, 2017].

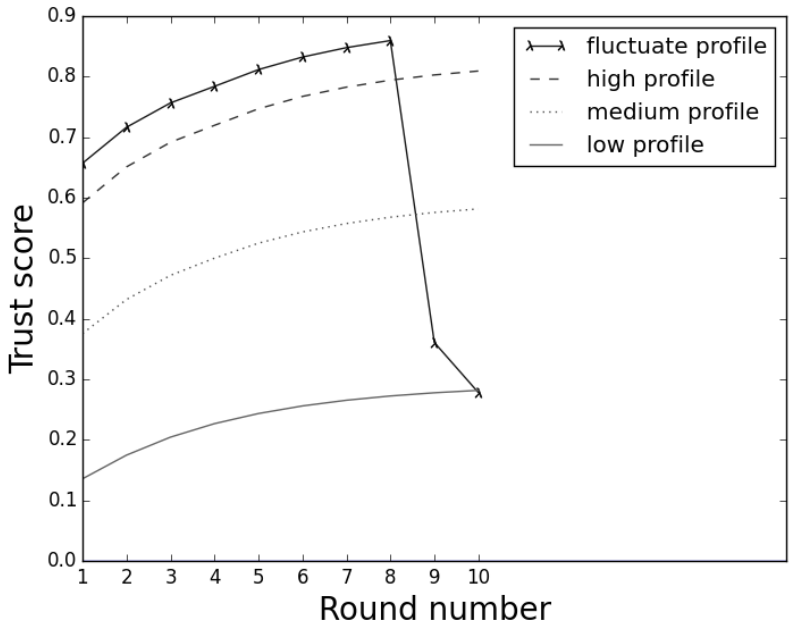


Figure 3.1: Trust scores calculated by our trust model for different user types in first 10 rounds.

Distinction between user types We analyze the behavior of our trust model on constant sending behavior versus fluctuating behavior. However, the constant sending behavior is not realistic. We relax our user profiles by allowing them to vary their behavior around the average sending amount. In particular, we define the behavior of *low profile*, *medium profile* and *high profile* as normal distributions with means of 0.2, 0.5 and 0.8 respectively, and standard deviation of 0.15 (this standard deviation value has been approximated from the meta analysis of 23,000 trust game players presented in [Johnson and Mislin, 2011]). In what follows we analyze whether our trust model can distinguish between different user types. Hence, after a large number of rounds, trust scores of different users will follow a distribution. In order to distinguish between different profiles, these distributions must satisfy the following properties:

- The trust scores assigned to *fluctuating* users should be similar with the trust scores assigned to *low profile* users, and should not overlap with the trust scores assigned to *medium profile* users.
- The difference between two mean values should be at least the sum of two standard deviations. If we denote by $mean_{low}$, $mean_{medium}$ and $mean_{high}$ the mean values of trust scores of *bad profile*, *medium profile* and *high profile* respectively, and by std_{low}, std_{medium} and std_{high} the corresponding standard deviations, then:

$$mean_{low} + std_{low} \leq mean_{medium} - std_{medium} \tag{3.12}$$

$$mean_{medium} + std_{medium} \leq mean_{high} - std_{high} \tag{3.13}$$

- The ratio of any two variances of these distributions should not be larger than 3, as suggested by [Keppel, 1991].

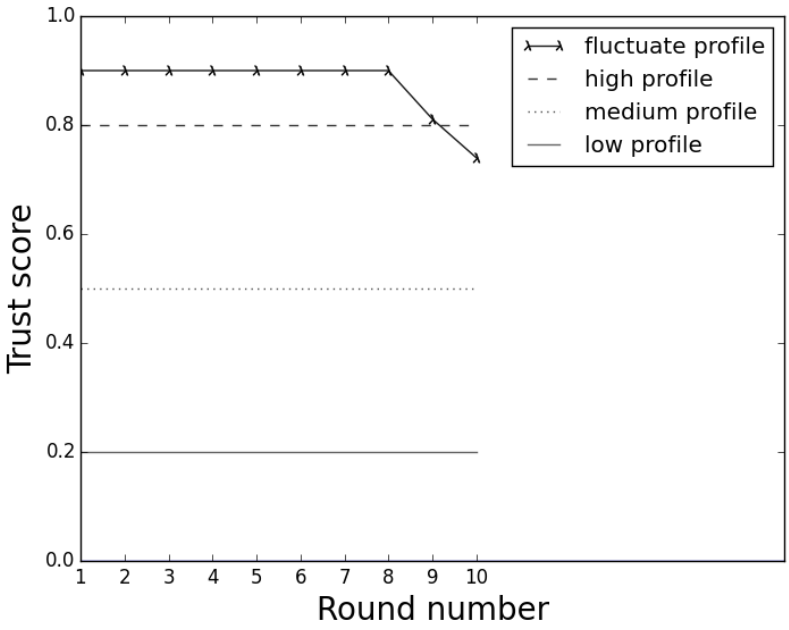


Figure 3.2: Trust scores calculated by the average model for different user types in first 10 rounds.

It is not easy to find a *current_trust* function which satisfies these above requirements. After an empirical process, the formula presented in Equation 3.2 is the only function we found so far that can satisfy these requirements. We select the initial values of parameters in Section 3.1.1.4 by using the same empirical process.

For instance, if we replace our *current_trust* formula by a new formula such as *current_trust* = *send_proportion*, this trust model will not be able to distinguish between *medium profile* and *fluctuating* users. As we show in Figure 3.3, after ten rounds, the new trust model will assign overlapping trust scores to *medium profile* and *fluctuating* users, but our trust model still can distinguish between these two user profiles as displayed in Figure 3.4.

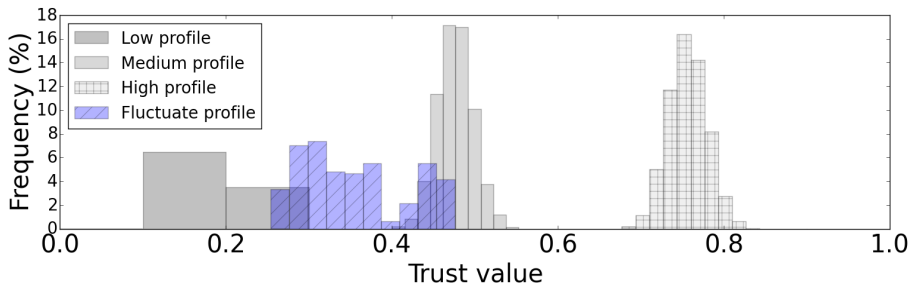


Figure 3.3: Distribution of the trust score calculated by the trust model with *current_trust* = *send_proportion* after ten rounds. The trust scores assigned to *fluctuating* users overlap with trust scores assigned to *medium profile* users.

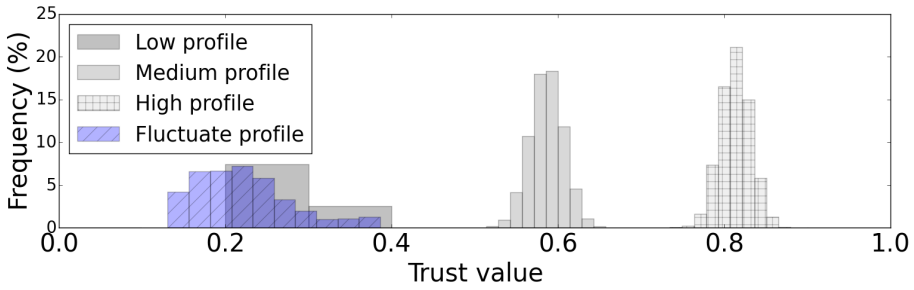


Figure 3.4: Distribution of our trust score calculated by our trust model after ten rounds. The trust scores assigned to *fluctuating* users do not overlap with trust scores assigned to *medium profile* users.

3.1.2.2 Consistency with human opinions

In this section, we evaluate our trust model according to user ratings obtained by an existing experimental study of the repeated trust game [Keser, 2002].

[Keser, 2002] organized a repeated trust game experiment where users could rate in each round their partners' sending behavior. The three levels proposed were: negative, neutral or positive. Based on the data published in this study, we created three virtual users called *positive user*, *neutral user* and *negative user* respectively corresponding to the levels of possible ratings. These virtual users follow the average behavior of real users who have the corresponding rating.

In what follows we analyze the results we obtained by applying our trust model to the behavior of these virtual users. Since we are using a continuous rating score and Keser was using a discrete rating score, the two rating scores do not match completely. However, we should expect that our trust model does not conflict with Keser's results, i.e. for any two behaviors A and B, if A was rated higher than B (for instance, positive versus neutral or positive versus negative), our trust model should assign a higher trust score to A than B.

The analysis is displayed in Figure 3.5. As expected, our trust model assigns in all cases higher trust values to *positive user* than *neutral user*, and higher trust values to *neutral user* than *negative user*.

The conclusion is that our trust model and people's opinion about trustworthiness of behavior in repeated trust games do not contradict each other.

3.1.2.3 Evaluation with real data

We showed that our trust model matches real people's opinions about partner's behavior in the past. In this section we address the issue whether it can predict the future behavior of users. For instance, if our trust model assigns a high trust value for a user, we are interested whether this particular user behaves well or badly in the future.

We note that, a low R-squared value is usual in predicting human behavior However in many cases, it does not mean that the prediction is useless [Gunnthorsdottir et al., 2002]. For instance, [Ashraf et al., 2006] used a list of ten factors to predict users' behavior in one-shot trust game, and achieved the average R-squared of 0.25.

Observation on data First, we show that our models on user profiles (*low, medium, high and fluctuate profiles*) are consistent with data collected throughout experiments. Next, we show that real data proves the existence of different user types such as participants who send

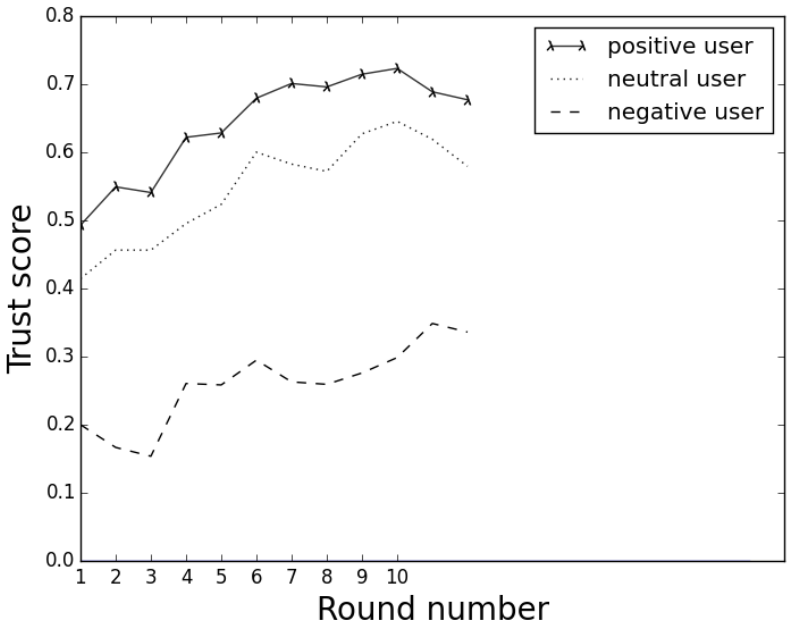


Figure 3.5: Validating trust model with real users' ratings.

in average a high amount and those who send in average a low amount. We also show that real data proves the existence of users with a fluctuating behavior.

We notice that changes in user behavior in repeated trust games are very usual. Figure 3.7 illustrates the average and standard deviation of sending amount proportions of each user in the three datasets previously mentioned. The standard deviations of user sending proportions are large compared with their average sending proportions, meaning that users often change their sending behavior during the experiments. For instance, Figure 3.6 illustrates a selected user behavior from our dataset: this player cooperates very well at beginning then deviates and never cooperates again. We observed that in all data sets, only few players send a constant amount throughout a session.

Figure 3.7 shows that for all three datasets, the average sending proportions of participants vary from 0 to 1, matching with our defined profiles: *low*, *medium* and *high* corresponding to a sending proportion of 0.2, 0.5 and 0.8 respectively.

We can conclude that, fluctuating behavior is a fact in all three data sets, and for this reason, it is important to design a trust model that copes with this behavior.

Predicting users' behavior In Section 2.3, we presented the trust analysis on user behavior in interaction with other information. Here we focus only on the predictive performance of trust function only. We also implemented a slightly different evaluation method. Instead of dividing sender and receiver role, we consider them together to see if the trust score can be used to predict users' behavior in both roles.

Based on the behavior log we applied our trust model on users' behavior at a certain round, then used the output trust score as the independent variable to predict the user's behavior in the next round. For all rounds starting with round five, we found a high correlation between the output trust scores and user behavior in the next round.

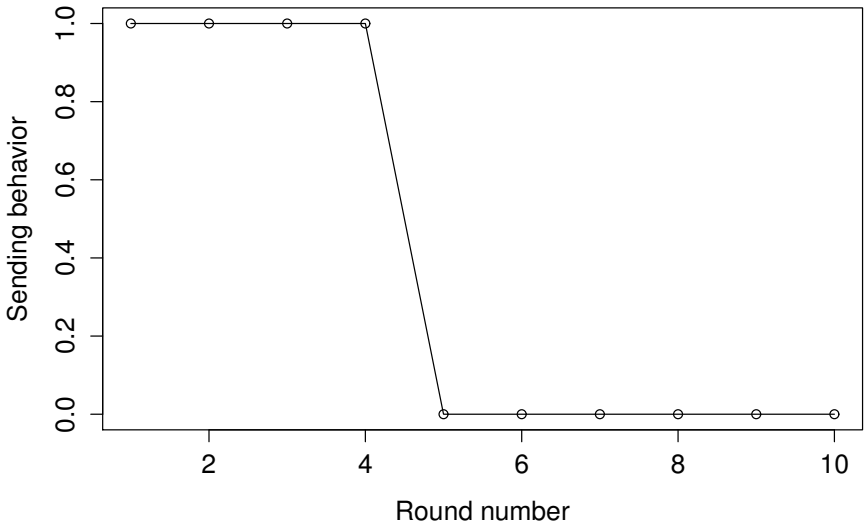


Figure 3.6: An observation of fluctuating behavior from our data set.

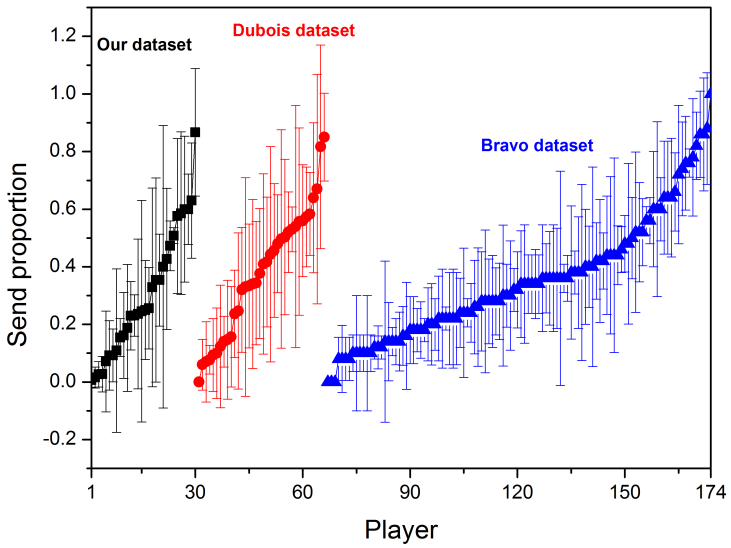


Figure 3.7: Average and standard deviation of sending proportions in datasets.

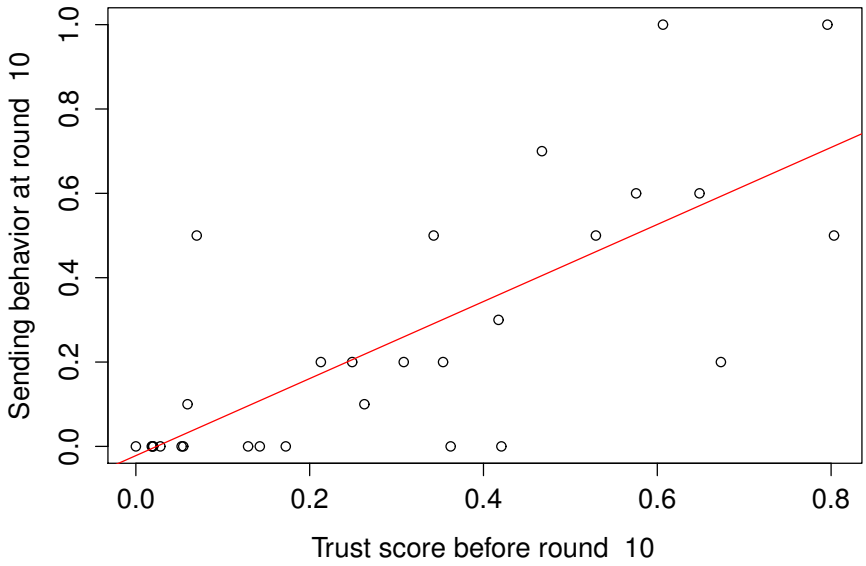


Figure 3.8: Relationship between trust score and user behavior at round ten in our own experiment.

In our analysis the independent variables are the trust score for each user after fourth and ninth interaction and the dependent variables are the sending proportions of users in the fifth and tenth round. For the data in [Dubois et al., 2012], we tested the relationship between our trust model and the user behavior at round five and ten. However, because of the design of the experiment in [Bravo et al., 2012], we could only test the relationship between our trust model and user behavior at round five. Figure 3.8 displays the prediction of user sending behavior at round ten by using the data set from our experiment. Figure 3.9 displays the prediction of user sending behavior at round five by using the Bravo dataset.

	Intercept	**Slope**	**Adjusted R-square**
Our dataset (round 5)	0.071	0.701***	0.319
Our dataset (round 10)	-0.022	0.913***	0.542
Bravo dataset (round 5)	-0.006	0.715***	0.362
Dubois dataset (round 5)	0.072	0.848***	0.356
Dubois dataset (round 6)	0.095	0.855***	0.451
Dubois dataset (round 7)	0.058	0.969***	0.414
Dubois dataset (round 8)	-0.007	1.027***	0.487
Dubois dataset (round 9)	0.049	0.878***	0.330
Dubois dataset (round 10)	0.027	0.855***	0.357

Table 3.2: Regression analysis of our trust function applied on external datasets. All slope values are significant at the level of 99.9%.

The summary of all linear regressions previously mentioned is displayed in Table 3.2, where

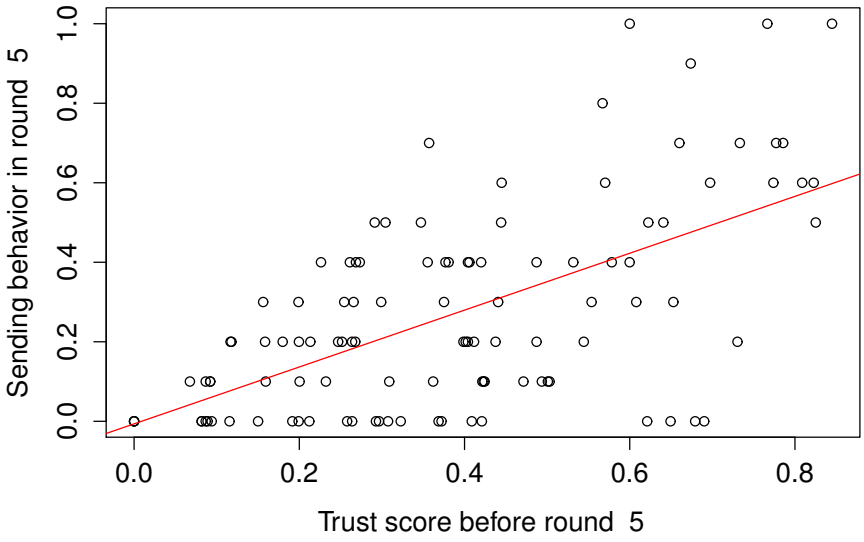


Figure 3.9: Relationship between trust score and user behavior at round five in the Bravo dataset.

the independent variable (x-axis in Figure 3.8 and Figure 3.9) is the trust score our model assigned to each user before a particular round, and the dependent variable is the behavior of this user in this round (y-axis in Figure 3.8 and Figure 3.9). We notice that the slopes of all regressions are significant, meaning that our trust metric predicts well user's behavior. Similar results were obtained for the same analysis in other rounds (i.e. a significant slope value and a positive r-value).

Comparison with baseline methods As previously mentioned, there is no prior work in predicting users behavior in repeated trust game. For this reason, in this section, we compare our model with two other baseline models: average model and null model.

Average model predicts that, the next sending amount of a user is equal to the average of her previous sending amounts. On the other hand, the null model predicts that, the next sending amount of a user is equal to her previous sending amount.

In order to compare the performance of these three models, we calculate the predicting values of each of these models. We compute the adjusted R-squared value for each model from round five to round ten and then calculate the average of adjusted R-squared. The higher average R-squared a model achieves, the better this model is in predicting users behavior.

The comparison of performance of different models is displayed in Table 3.3. For our data and data of Dubois, we calculate an average adjusted R-squared values in predicting users behavior from round five to ten. As Bravo's dataset contains only five rounds, we computed the average adjusted R-squared values in predicting users behavior at round five.

We can see that, our model outperforms the other two baseline models in predicting users behavior in repeated trust games.

In this section, we presented our computational trust model for repeated trust games. To the best of our knowledge, it is the first trust model for repeated trust game which has been

Table 3.3: Comparison of R-squared values of different predicting models.

	Average model	**Null model**	**Our model**
Our data	0.42	0.43	0.55
Dubois's data	0.28	0.34	0.40
Bravo's data	0.3	0.32	0.36

presented. We validated the trust model against (i) simulated data to verify the model on predefined user behavior patterns, (ii) human opinions to verify that the model is consistent with human idea about trust level of partners, and (iii) real user behavior datasets collected from different lab experiments to verify the prediction power of the model on user behavior.

The trust model is based on the assumption that we know the numerical values of each behavior of users in the context. It is easy in trust game context because user behaviors are already represented by numerical values. However, in order to extend the trust model into other contexts we need to find a way to define these values in these contexts. In the next section, we will present an application of our computational trust model in Wikipedia. First of all we will present several methods to automatically assess the quality of Wikipedia articles, then based on the quality of Wikipedia articles we can measure the contribution of each Wikipedia editor. The experimental results showed that by applying our computational trust model on Wikipedia we can better predict the future contribution of Wikipedia editors compared to the predictions made by baseline models.

3.2 Trust Calculation in Wikipedia

In the previous section we presented a computational trust model for repeated trust game which is based on the goodness of behavior of users in the past. As we discussed, it is easy to measure the *goodness* of user behavior in trust game because they are represented by numbers. In this section, we will focus on a real-world collaborative system which is Wikipedia. The task of defining the quality of previous behavior of users in Wikipedia is not an easy task and some research works need to be done to automatically assess the quality of contribution of users before we can design a trust model for Wikipedians.

3.2.1 Why Wikipedia?

Wikipedia is considered as the largest knowledge repository that has been created through the human history. At the time of writing, Wikipedia contains more than 41 millions articles in all languages with 5.3 millions articles particularly belong to English Wikipedia, as the result of the contribution from around 29 millions users[17]. The size of Wikipedia, in term of number of articles, has increased continuously since the beginning of Wikipedia in 2001 as displayed in Figure 3.10[18]. Moreover, Wikipedia is being modified in an impressive speed. According to Wikipedia Statistics[19], on average per second ten edits are performed on Wikipedia.

Wikipedia is a dominant information source for the entire generation of Internet users [Brown, 2011]. Present day users tend to take for granted from Wikipedia, even for a serious and

[17] https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia
[18] Image source: https://en.wikipedia.org/wiki/Wikipedia:Modelling_Wikipedia's_growth
[19] https://en.wikipedia.org/wiki/Wikipedia:Statistics

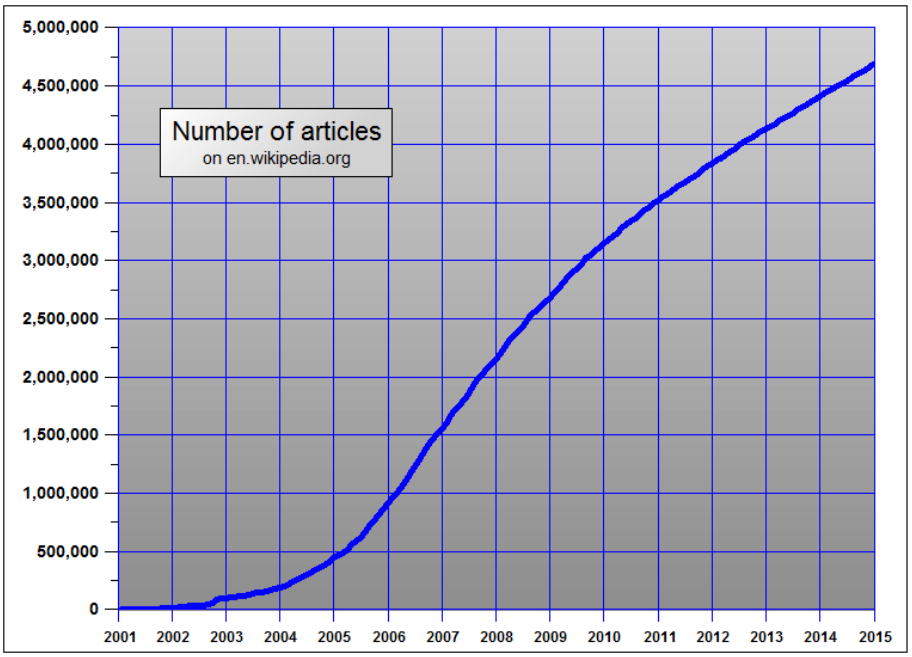


Figure 3.10: Number of English Wikipedia articles from 2001 to 2015.

dangerous problem such as health-care information [Jones, 2009]. This phenomenon is more popular among youths [Pan et al., 2007; Rowlands et al., 2008]. Furthermore, users of popular search engines such as Google usually reach to Wikipedia [Natalie Kupferberg et al., 2011] because Wikipedia is usually selected among the first result of Google search queries. As a result, the influence of the information presented on Wikipedia is increased. It is reported that the physicists might try to take the information from Wikipedia for their works [Pinsker, 2015].

A study [Goodwin, 2012] looked at 1,000 search terms in Google and measured the rankings for the website `Wikipedia.org`. The study found that Wikipedia is page one of Google for 99% of searches (of nouns), that Wikipedia is position one of Google for 56% of searches and that 96% of searches had Wikipedia in position 1-5 on Google. Today, a query on Google usually returns directly the content of corresponding Wikipedia page as showed in Figure 3.11.

Wikipedia is very well-organized website. A lot of annotated Wikipedia data is available. The Wikipedia dataset makes it easier to analyze and study in comparison with free text platforms such as Google Docs. We could expect that, studies on Wikipedia will be the first step in studying real-world collaborative systems.

3.2.1.1 Trust between Wikipedians

Trust is a very important factor in collaborative editing activities in general and Wikipedia in particular. When an author collaborate with other authors to write an article, she needs to decide to trust and collaborate with these partners or not, or should she grant some access rights to a particular coauthor or not.

As we will discuss in more details in Section 3.2.3.2, there is no previous work in measuring trust of Wikipedians. Existing studies that focus on measuring reputation are mostly based on quantitative metrics such as the number of edits made by an author, but have not discussed about the qualitative metrics such as the quality of the text.

We present a computational trust model for Wikipedia authors. The main idea of the

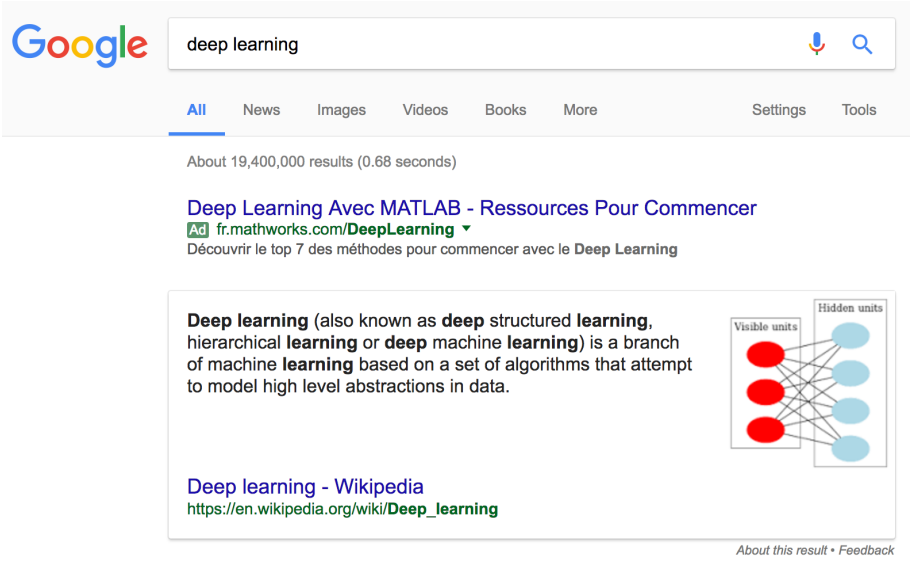


Figure 3.11: Google displayed the content from Wikipedia as the search result.

computational trust model is, a coauthor who collaborated before in high quality text is expected to produce high quality text in the future, and therefore should be assigned a high trust score. However, different from trust game, it is not trivial to determine the quality of the contributions of users in Wikipedia. In this thesis, we propose to measure the quality of the contributions of users by the quality of the articles that the users contributed to. Therefore, in order to measure the trust between Wikipedians we need to know the quality of Wikipedia articles.

In fact, the quality of Wikipedia articles are being classified manually by reviewers [Wikipedia, 2017g] with the support of some automatic tools such as ORES [Foundation, 2015]. In the next section we will discuss why does this approach not work well and the reason why we need an automatic solution for quality assessment.

3.2.2 Problem Definition

As discussed above, in order to measure the trust score of users we need to know the quality of Wikipedia articles in which they are involved. Therefore the problem of measuring trust of Wikipedia authors is divided into two sub-problems. First we need to find a method to measure the quality of Wikipedia articles. Secondly we need to define a trust metric which takes the quality of collaborative articles into account.

3.2.2.1 Quality of Wikipedia articles

We define the problem as follow. For a given language of Wikipedia, a set of quality class is defined. A set of Wikipedia articles (represented with revision ID) whose quality classes are already assigned by human reviewers is provided as *ground truth* [Shalev-Shwartz and Ben-David, 2014]. We need to design an algorithm to predict the quality class of a new Wikipedia article. This is a multi-class classification problem [Shalev-Shwartz and Ben-David, 2014].

The available quality classes for each English, French and Russian Wikipedia datasets are provided as below. The definition and requirements for each quality classes in English Wikipedia are provided in Table 3.4 [Wikipedia, 2017b]. Similar definitions for French and Russian Wikipedia can be found on corresponding Wikipedia language sites.

English : FA, GA, B, C, Start, Stub.

French : ADQ, BA, A, B, BD, E.

Russian : FA, GA, SA, I, II, III, IV.

Class	Description
FA	Professional, outstanding, and thorough; a definitive source for encyclopedic information.
GA	Useful to nearly all readers, with no obvious problems; approaching (but not equalling) the quality of a professional encyclopedia.
B	Readers are not left wanting, although the content may not be complete enough to satisfy a serious student or researcher.
C	Useful to a casual reader, but would not provide a complete picture for even a moderately detailed study.
Start	Provides some meaningful content, but most readers will need more.
Stub	Provides very little meaningful content; may be little more than a dictionary definition. Readers probably see insufficiently developed features of the topic and may not see how the features of the topic are significant.

Table 3.4: Description of English Wikipedia's quality labels

Currently, human reviewers are reviewing and assigning quality classes to Wikipedia articles manually [Wikipedia, 2017g]. The reviewers might need to review an article after each modification on the article, because the quality might change dramatically after even a single modification. However, due to the very high modification speed of Wikipedia, human resources are simply not enough to review every Wikipedia revisions. We need an automatic solution. As a matter of fact, ORES service [Halfaker and Taraborelli, 2015] has been used since 2014 to assist Wikipedia users in determining the quality of Wikipedia articles. ORES is built based on the work of [Warncke-Wang, Ayukaev, et al., 2015]. In this thesis we will consider these works as state-of-the-art.

Studies [Warncke-Wang, Ayukaev, et al., 2015; Blumenstock, 2008; Suzuki, 2015; Betancourt et al., 2016] proposed different approaches to assess the quality of Wikipedia articles. Generally speaking, most existing approaches are based on traditional machine learning algorithms such as *svm* or *random forest*. The common characteristic of these algorithms is they all require their input as a manual designed feature set.

A feature is defined as an individual measurable property of the process being observed [Chandrashekar and Sahin, 2014]. We can consider a feature set as a simplified model of the Wikipedia articles. Designing a good feature set is a very difficult task in machine learning [Ng, 2013]. In addition, the feature set is usually designed for a specific task and does not generalize well. For instance, measuring quality of Wikipedia articles in different languages require different feature sets [Halfaker and Taraborelli, 2015].

We propose three different approaches in assessing the quality of Wikipedia articles. The first approach is an extension of state-of-the-art [Warncke-Wang, Ayukaev, et al., 2015] by adding more features. In the second approach, we use Doc2Vec [Le and Mikolov, 2014] to convert articles into vectors then used Deep Neural Network to predict the quality labels of articles. In the third approach, we use Recurrent Neural Network (RNN) [Rumerhart et al., 1986] with

Long-Short Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] to build an end-to-end classifier.

We can use the second and the third approach in any language while the first approach is available only for English. In term of accuracy and AUC the third approach achieves the highest performance, but the running time of the first approach is very low (in order of seconds) compare to other two approaches (the running time is in order of hours). Therefore, we can use the first approach if the computational resource is limit, while the third approach is suitable if the task is not needed to be finished immediately.

We note that, the problem we are considering is measuring the quality of a given Wikipedia article, i.e. to measure how *well* an article is written. The task is not to measure the correctness of the information presented in the article, i.e. how *true* an article is written [Y. Zheng et al., 2017].

3.2.2.2 Measuring Trust of Wikipedia Coauthors

We define the problem as follows.

Given an author whose name is Alice in Wikipedia. Alice has joined Wikipedia for a while and collaborated with n partners in total so far denoted by p_1, p_2, .. p_n in m collaborative articles denoted by a_1, a_2, ... a_n.

Indeed Alice collaborated with different partners in different documents. For instance, she collaborated with p_1 and p_3 in the article a_1, but collaborated with the partners p_3 , p_4 and p_5 in the article a_2.

Now, given the modification log of all articles that involved Alice, we need to assign a trust value for each partner of Alice.

In the next section we will review related studies to the two problems that are measuring quality of Wikipedia articles and measuring trust of Wikipedia editors.

3.2.3 Related Work

3.2.3.1 Measuring Quality of Wikipedia

Existing studies in measuring quality of Wikipedia articles rely on traditional machine learning with hand-engineered features. Based on the nature of the features that are used, we can roughly divide them into two categories: *editor-based* and *article-based* features.

Editor-based features Approaches that used editor-based information analyzed information that cannot be computed uniquely from the current content of Wikipedia pages, such as the authors of a particular article, their contributions and the duration of each contribution.

Using the hypothesis that the more reputable an author is, the higher the quality of the articles this author produces, Hu et at. [M. Hu et al., 2007] and Adler et al. [B. Thomas Adler, Chatterjee, et al., 2008] used *reputation* of authors to determine the quality of Wikipedia articles. The result was confirmed in German Wikipedia [K. Stein and Hess, 2007]. The social capital of the editors could also affect the quality of the articles they contributed [Nemoto et al., 2011]. Using statistical approach, Javanmardi and Lopes [Javanmardi and C. Lopes, 2010] verified that the editors reputation can be used to detect the quality of Wikipedia articles. Suzuki applied the *h-index* on academic ranking for assessing the quality of an article [Suzuki, 2015]. Li et al. [Xinyi Li et al., 2015] presented a modified weighted PageRank algorithm in the network of editors and articles to assess Wikipedia quality. [Betancourt et al., 2016] studied the team characteristics, such as how many FA or GA articles the team members have worked on before, to predict the

quality class of Wikipedia articles. The authors limit their work at classifying FA-GA articles with other articles, but not classify all quality classes as the work of [Warncke-Wang, Ayukaev, et al., 2015].

Another criteria used for assessing the quality of a text is the period of time the text remains stable or is modified by other authors/reviewers. If an article has not been modified significantly for a long time, this article can be considered as mature and of high quality. For instance, [Calzada and Dekhtyar, 2010] presented the idea of *stable* articles to determine the quality of Wikipedia. [Wöhner and R. Peters, 2009] also claimed that a good article should not be modified for a long enough period of time.

Some other works presented the idea that the quality of Wikipedia articles can be determined based on the interaction between authors and reviewers [La Robertie et al., 2015; G. Wu et al., 2012]. [Wilkinson and Huberman, 2007] showed that a large number of authors and reviewers with an intensive cooperation should lead to high quality articles. [Kittur and R. E. Kraut, 2008] showed that the effectiveness of adding contributors is dependent on the degree and type of coordination those contributors use. [Arazy and Nov, 2010] showed that inequality of editors' contribution in a particular article, and inequality in overall Wikipedia activity levels for the same set of editors affect document quality. Liu and Ram [J. Liu and Ram, 2011] suggested that the behavior pattern of editors also effects articles quality.

Article-based features The second main approach of assessing quality of Wikipedia articles is to analyze directly the content of Wikipedia articles.

One of the simplest solutions is to measure the length of Wikipedia articles [Blumenstock, 2008]. This solution achieved a very high accuracy in separating between *FA* and *non-FA* articles. With the same target to distinguish between *FA* and *non-FA* articles, other works considered the writing styles, such as how editors vary the words they used, for assessing the quality of articles [Lipka and B. Stein, 2010; Xu and Luo, 2011].

Dalip et al. [Dalip, Goncalves, et al., 2009] analyzed the effect of the feature set comprising text, review and network on the quality of Wikipedia articles. The authors verified the correlation between this feature set and the quality of Wikipedia articles. They claimed that, using the error term of linear regression, the features that describe the structure and style of the articles are the best to distinguish between articles of different quality classes.

Similarly, using content, structure and network and edit history features, Anderka et al. [Anderka et al., 2012] built a binary classifier to predict quality flaws in Wikipedia. They based their approach on the *cleanup tags*, which are given by the reviewers who detected the flaws but do not have enough time or expertise to fix them.

Focusing on the feature set that describes the content of the Wikipedia articles, Warncke-Wang et al. [Warncke-Wang, Cosley, et al., 2013] presented and analyzed the feature set including 17 features, such as article lengths and the number of headings of an article. Authors claimed that there are 11 features that should be considered to evaluate the quality of Wikipedia articles. The result is presented in [Warncke-Wang, Ayukaev, et al., 2015].

Based on the work of [Warncke-Wang, Ayukaev, et al., 2015; Warncke-Wang, Cosley, et al., 2013], Wikimedia Foundation[20] built an online service to predict the quality class of Wikipedia articles called ORES (*Objective Revision Evaluation Service*) [Halfaker and Taraborelli, 2015]. Currently, ORES provides predicting services for English, French and Russian Wikipedia. For each language, a new feature set is required, though some features are shared[21].

[20] https://wikimediafoundation.org

[21] The work of ORES has not been published anywhere but only on the Wikipedia website and is subject to be

Other Related Studies The research works we mentioned above directly study the problem of measuring quality of Wikipedia articles. There are other research studies which are not directly solving the problem, but focus on very close issues.

Instead of measuring the quality of Wikipedia articles, [Suzuki and Nakamura, 2016] tried to measure the quality of Wikipedia editors. Interestingly, the authors proposed to use a crowd-sourcing system to address the issue related to a particular revision, i.e. the authors used some crowd-sourcing platforms like Amazon Mechanical Turk [N. Zhang, 2010] to recruit workers to analyze an article. The analysis result is accepted only if two workers agree with each other.

Another related task to measuring quality of Wikipedia articles is to detect vandalism in Wikipedia revision [Potthast et al., 2008; Tramullas et al., 2016]. An example of a Wikipedia vandalism is displayed in Figure 3.12. As discussed above, the task of measuring quality of Wikipedia articles does not aim to measure the trustworthiness of the information given by the authors. For instance, the two sentences "Napoleon is a man." and "Napoleon is a girl." are considered as similar in the view of a quality measuring algorithm. Therefore, before measuring the quality of an article, we need to detect and remove any vandalism to ensure that the information is correct at a certain level [Halfaker, Kittur, et al., 2011]. As of this writing, vandalism is effectively detected in Wikipedia by using both manual and automatic ways [Wikipedia, 2017d].

Figure 3.12: An example of Wikipedia vandalism. In fact, as of this writing, Jason Terry is still alive.

3.2.3.2 Measuring Trust of Wikipedia Authors

To the best of our knowledge, there is no prior research work on assigning trust scores to Wikipedia authors. However, several studies on measuring reputation of Wikipedians are presented[22].

[B. Thomas Adler and Alfaro, 2007] presented a *content-driven* reputation scheme. The authors used *longevity* of text to measure the reputation of Wikipedia editors, i.e. an author

updated. The results we presented in this thesis are based on the information retrieved from [Wikimedia, 2016].

[22]Several authors used the term *trust* [Javanmardi, Ganjisaffar, et al., 2009], but in fact they refer to *reputation* of authors.

gains her reputation if their modification is not modified by subsequent authors. Naturally, the reputation of an author fails if their modification is removed by other ones. The ideas then are implemented in WikiTrust [B. Thomas Adler, Chatterjee, et al., 2008; Alfaro et al., 2011; B Thomas Adler, 2012]. However, as of this writing WikiTrust service has been shut down [WikiTrust, 2017]. The idea of [B. Thomas Adler and Alfaro, 2007] is extended in [Javanmardi, C. V. Lopes, et al., 2010], where the authors consider not only the inserting text, but also other activities such as deleting or rolling back to measure the contribution of authors.

[Meo et al., 2017] studied a related issue with our study, but on a reverse direction. Starting from RfA trust networks when Wikipedia users declares that they *trust* or *distrust* other users [Burke and R. E. Kraut, 2008], the authors tried to predict the reputation the network members.

In this section, we presented the state-of-the-art in two different problems: measuring quality of Wikipedia articles and measuring the reputation of Wikipedia editors. In the next two sections we will present our approaches for these two problems.

3.2.4 Measuring Quality of Wikipedia Articles

We present three different approaches to measure the quality of Wikipedia articles. In the first approach, we improve the state-of-the-art model [Warncke-Wang, Ayukaev, et al., 2015] by adding new hand-designed features. In two other methods, we present novel techniques using deep learning for automatic feature extraction and end-to-end learning for quality measurement.

3.2.4.1 Improvement of existing studies

State-of-the-art As of this writing, the model presented in [Warncke-Wang, Ayukaev, et al., 2015] is considered as state-of-the-art. [Warncke-Wang, Ayukaev, et al., 2015] analyzed English Wikipedia articles and defined 11 features to represent the quality of a Wikipedia article. The list of features are as follow. The names inside the parentheses are the corresponding variable names. We use the variable names later to represent the features.

- Article length in bytes (*content_length*)
- Number of references (*num_references*)
- Number of outlinks to other Wikipedia pages (*num_page_links*)
- Number of citation templates (*num_cite_temp*)
- Number of non-citation templates (*num_non_cite_templates*)
- Number of categories linked in the text (*num_categories*)
- Number of images / length of article (*num_images_length*)
- Information noise score (*info_noise_score*) [Stvilia et al., 2008]
- Article has an infobox or not (*has_infobox*)
- Number of level 2 headings (*num_lv2_headings*)
- Number of level 3+ headings (*num_lv3_headings*)

Variable	Formula
avg_sentence_len	$\frac{number\_of\_words}{number\_of\_sentences}$
avg_word_len	$\frac{number\_of\_letters}{number\_of\_words}$
avg_syllables_per_word	$\frac{number\_of\_syllables}{number\_of\_words}$
percentage_of_difficult_words	$\frac{number\_of\_difficult\_words}{number\_of\_words}\%$

Table 3.5: Definition of variables used in readability scores

Adding features The above feature list does not take into account how the articles are written. Other studies [Lipka and B. Stein, 2010] claimed that writing style does matter in assessing the quality of Wikipedia articles. Therefore, we improved the model by adding nine readability scores into the feature set, so in total we have a feature set of 20 features.

The list of added features are:

Flesch reading score (*flesch_reading_ease*) Flesch reading score, or Flesch reading ease [Kincaid et al., 1975], is a measure to test how difficult to understand an English text. Flesch reading ease for a given text is a number between 100 and 0, where higher scores indicate text that is easier to read while lower numbers mark text that is more difficult to read.

$$flesch\_reading\_ease = 206.835 - (1.015 \times avg\_sentence\_len) - (84.6 \times avg\_syllables\_per\_word) \tag{3.14}$$

Flesch-Kincaid grade level (*flesch_kincaid_grade*) Flesch-Kincaid grade level [Kincaid et al., 1975] for a given English text is a number corresponding to the US grade level required to understand the text. For example, if the score is 9.3, it means that the reader of the text should be ninth grader or higher. Although Flesch reading ease and Flesch-Kincaid grade level use both word length and sentence length as core measures, they have different weighting factors. These measures are inversely correlated: a text with a high score on the reading ease test should have a low score on the grade-level test.

$$\begin{aligned} flesch\_kincaid\_grade = {} & 11.8 \times avg\_syllables\_per\_word \\ & + 0.39 \times avg\_sentence\_len - 15.59 \end{aligned} \tag{3.15}$$

Smog index (*smog_index*) Smog index [McLaughlin, 1969] of a text estimates the years of education a person needs to understand a given text in English.

$$smog\_index = 3 + \sqrt{polysyllable\_count} \tag{3.16}$$

The *polysyllable_count* is defined as the number of words with more than two syllables.

Coleman-Liau index (*coleman_liau_index*) Coleman-Liau index, or Coleman-Liau readability formula [Coleman and Liau, 1975] is a linguistic test that measures as Flesch-Kincaid grade the US grade level thought necessary to comprehend a text. As opposed to Flesch-Kincaid grade, Coleman - Liau index relies on characters instead of syllables per word.

$$\begin{aligned} coleman\_liau\_index = 5.88 \times avg\_word\_len \\ -29.6 \times avg\_sentence\_len - 15.8 \end{aligned} \tag{3.17}$$

Automated readability index (*automated_readability_index*) Automated readability index (ARI) [Senter and E. Smith, 1967] is another readability score to detect the readability of a given text in English in terms of the US grade level similar to Flesch-Kincaid grade and Coleman - Liau index. ARI and Coleman-Liau index rely on a factor of characters per word, instead of syllables per word as the other listed measures.

$$\begin{aligned} automated\_readability\_index = 4.71 \times avg\_word\_len \\ +0.5 \times avg\_sentence\_len - 21.43 \end{aligned} \tag{3.18}$$

Difficult words (*difficult_words*) The difficult words score [Jeanne Sternlicht Chall and Dale, 1995] of a given English text is calculated based on how many difficult words appear in a text. A word is considered difficult if it does not appear in a list of 3000 common English words that groups of fourth-grade American students could reliably understand.

Dale-Chall score (*dale_chall_readability_score*) Dale-Chall readability score [Dale and Jeanne S Chall, 1948] is another measure for comprehension difficulty when reading a text. This score takes into account the percentage of difficult words in the text as well as the ratio between the number of words and the number of sentences.

$$\begin{aligned} dale\_chall\_readability\_score = 0.1579 \times percentage\_of\_difficult\_words \\ +0.0496 \times avg\_sentence\_len \end{aligned} \tag{3.19}$$

Linsear write formula (*linsear_write_formula*) Linsear Write Formula is a readability score initially designed for the United States Air Force to compute the readability of their technical manuals [H. Chen, 2012]. This score corresponds to the US grade level of a text sample based on sentence length and the number of words used that have three or more syllables.

More precisely, based on a sample of 100 words from the text, where the number of words with two syllables or less is denoted by n_1 and the number of words with three syllables or more by n_2, Linsear Write Formula is calculated as $\frac{n_1+3\times n_2}{number\_of\_sentences\times 2}$ if $\frac{n_1+3\times n_2}{number\_of\_sentences} > 20$ and as $\frac{n_1+3\times n_2}{number\_of\_sentences\times 2} - 1$ in other cases.

Gunning-Fog index (*gunning_fog*) Gunning-Fog index [Gunning, 1969] is another readability score to measure the difficulty of a given text in terms of the years of formal education needed to understand the text on a first reading. It is a weighted average of the number of words per sentence, and the number of long words per word.

$$gunning\_fog = 0.4 \times (avg\_sentence\_len + percentage\_of\_difficult\_words) \tag{3.20}$$

After defining the set of 20 above features, we wrote a script to extract these features from Wikipedia articles, then feed the output vectors into *random forest* model for training and testing.

3.2.4.2 Novel approaches with deep learning

Issues of hand-engineered features Defining feature set is a very difficult task in machine learning [Chandrashekar and Sahin, 2014; Shalev-Shwartz and Ben-David, 2014]. A feature set can be considered as a simplified model of the given data, such as we model a person by height, weight, date of birth, etc. The key issue of manual feature engineering approach is information loss, i.e. there are always some missing information that are present in the raw data but are not available in the feature set. Usually this information is considered as irrelevant by the researchers [Chandrashekar and Sahin, 2014], but in fact we never know if these features are irrelevant or not, because they are never taken into consideration.

The information loss problem can be avoided if and only if the entire data is used as the feature set, as Norbert Wiener said, “the best material model of a cat is another, or preferably the same, cat" [Rosenblueth and Wiener, 1945].

Furthermore, feature engineering mostly relies on expert knowledge, which is usually expensive. Feature engineering requires effort and time, and when the researchers switch to a new problem, a new feature set is needed. Each Wikipedia language requires a new feature set to be designed, and it is difficult to do so without at least some basic understanding of this language. For instance, it is almost impossible to remove stop words, i.e. words with not many meanings like “a", “an", “the" in English, for Vietnamese Wikipedia without some knowledge about Vietnamese language and Vietnamese processing.

Many feature selection algorithms have been proposed [Chandrashekar and Sahin, 2014]. Their inputs are a large feature set with a lot of features and these algorithms try to remove irrelevant features. There is no automatic method to define the initial feature set - the best practical way is to add as many feature as possible. Particularly, [Aphinyanaphongs et al., 2014] performed a comprehensive analysis on feature selection for text categorization. In fact the authors suggested that using all features “consistently produces high and the nominally best *AUC* performance for the majority of classifiers". The work of [Aphinyanaphongs et al., 2014] suggested that, using the entire document contents for classification might be a good idea.

Someone could argue that, we can continue the traditional manual feature engineering approach by using feature selection methods. We can start with a complete feature set that contains all the possible features, then eliminate them one by one. Unfortunately, this approach is not feasible, not only because of computational resource requirements but also because the number of possible features we can extract from the raw data basically is infinitive. The reason is, along with primitive features which can be extracted directly from the raw data, we can also create new features based on primitive features. For instance, based on two primitive features which are the length of the document and the number of sentences in the document, we can create a new feature which is the average length of the sentences in the document. Definitely there is no limit in creating new feature by this way.

Deep learning [LeCun et al., 2015] can avoid manual feature engineering by learning directly from raw data. Furthermore, deep learning techniques take the entire Wikipedia articles as the input, hence it does not lose any information. In other words, the input of the deep learning algorithms is the same as the input of human reviewers. Therefore, theoretically an algorithm which can achieve the same assessment with human reviewers is possible, as proved in recent studies in different fields [J. S. Chung et al., 2016].

Automatic feature engineering with Doc2Vec Most machine learning algorithms including neural networks require the input to be represented as a fixed-length feature vector. As Wikipedia articles have different lengths, we need an approach that maps Wikipedia articles to fixed-length feature vectors. The most common fixed-length vector representation for documents is the *bag-of-words* [Harris, 1954] where a document is represented as the bag of its words. However, this approach disregards semantics and even word order.

As Wikipedia articles have various length, the classification task is more complex than in the case of fixed length articles. A common approach is by using *bag of words* to represent a document. However, this approach cannot capture the structure of a document, which might lead to ambiguities. For instance, *bag of words* cannot distinguish the two pieces of text "not good" and "good not" since they have the same words but in different orders.

In this thesis, we applied the unsupervised learning algorithm called *Paragraph Vector*, recently known as *Doc2Vec* [Le and Mikolov, 2014] that learns vector representations for variable-length pieces of texts and overcomes the disadvantages of *bag-of-words* by taking into account the order and semantics of words. In this approach every word and every paragraph are mapped to a unique vector. The paragraph vector is concatenated with several word vectors from the paragraph and trained in order to predict the next word in a text window. While word vectors are shared among paragraphs, paragraph vectors are unique among paragraphs.

The idea of *Doc2Vec* is to capture not only context around a word as the previous technique *Word2Vec* [Mikolov et al., 2013] does, but also to capture the order of the words in the document, which is an important factor in understanding the document, as displayed in Figure 3.14. An example of *Word2Vec* framework is displayed in Figure 3.13 where the model uses three words "the", "cat", and "sat" to predict the next word "on".

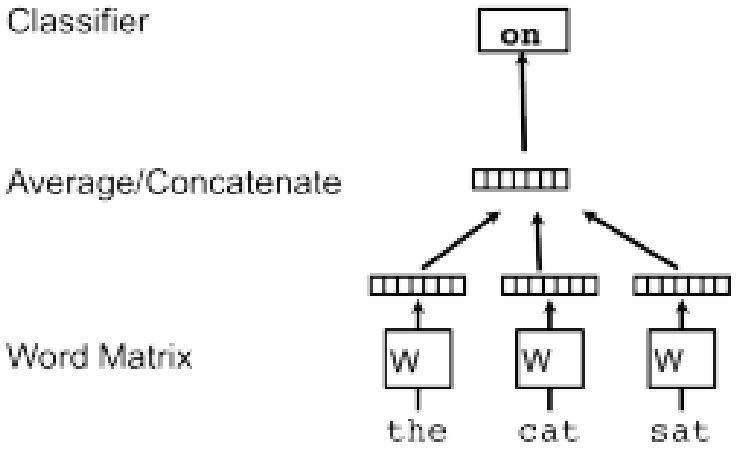


Figure 3.13: *Word2Vec* [Le and Mikolov, 2014].

We applied the *Doc2Vec* approach where each Wikipedia document corresponds to a paragraph in the above description. While the generated word vectors are not further used, the document vector is given as input for our deep neural network. The output vectors will be used as the input for a deep neural network to classify the quality class.

In implementation phase, we performed Doc2Vec with the $output\_size = 500$. We fed the output vectors in a DNN with four layers. Each layer of the DNN has 2000, 1000, 500 and 200 neurons respectively. We applied early stopping criteria in 5 epochs.

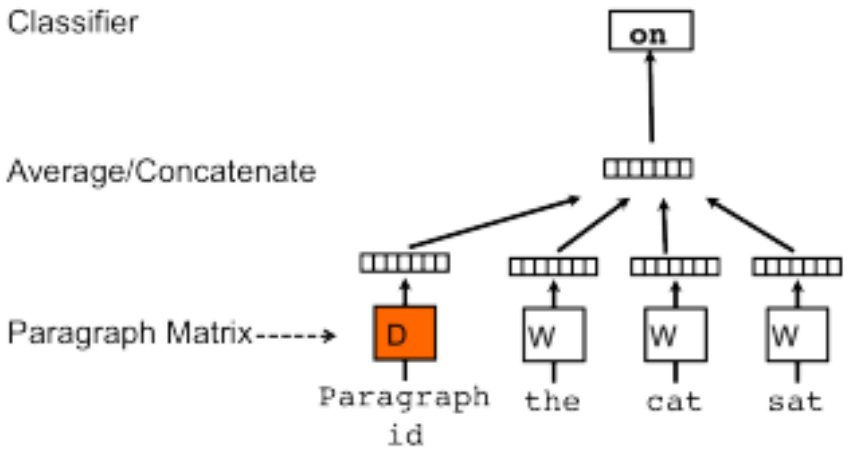


Figure 3.14: *Doc2Vec* [Le and Mikolov, 2014]. *Doc2Vec* is very similar with *Word2Vec*, except that it takes the document's content into account.

An end-to-end learning solution The approach of using *Doc2Vec* is promising. However, there are several disadvantages related to this approach:

- The result is not as good as other techniques.
- The model need to be retrained every-time when a new article arrives[23].
- The word can be divided into two independent phases with no information exchange between them. Therefore, the work limits the potential of weight sharing [Goodfellow et al., 2016].

We present an end-to-end learning method to predict the quality class of Wikipedia articles by using RNN with bidirectional LSTM. RNN and LSTM are presented in more details in Section A.2.2.2. In short, RNN is a neural network model that learns the data in sequence. RNN is a powerful tool to use in natural language processing, because the order of words is important in natural language.

The model is visualized in Figure 3.15 which can learn directly from the data input to the predicting output. The model is constructed with one embedding layer ($size = 300$), two stacked LSTM layers with 512 neurons of each layer, and finally a fully connected layer ($size = 6$ as the number of quality classes). Similar with [Gal and Ghahramani, 2016] we used Adam optimizer [Kingma and J. Ba, 2014] with dropout ratio [Zaremba et al., 2015] of 0.75, based on the studies of [Molchanov et al., 2017; C. Zhang et al., 2017] stated that a deep neural network is redundant enough for aggressive dropout values. Similar with [Gal and Ghahramani, 2016] we used *adaptive learning rate* with Adam optimizer [Kingma and J. Ba, 2014]. The initial learning rate is set at 0.001. We used the *batch_size* of 32. We set the number of training epochs as 200 but in fact the model became stable after around 100 epochs. All hyper-parameters are selected by using Random Discrete Search [Bergstra and Bengio, 2012].

In this section we presented three different approaches to automatically assess the quality of Wikipedia articles. In the next section, we will discuss the problem that given the quality of

[23]It took several day for training Doc2Vec model on 30,000 Wikipedia articles using a cluster with 2x8-core CPU and 250GB of memory.

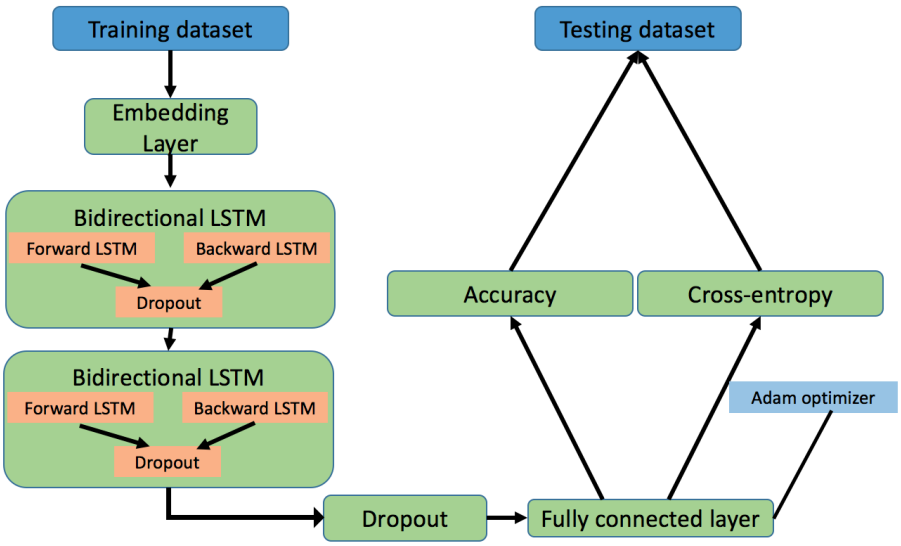


Figure 3.15: The bidirectional LSTM model to predict quality class of Wikipedia articles.

Wikipedia articles resulting from the contribution of a user and a partner, how can we calculate the trust score of the partner for the user.

3.2.5 Measuring trust of coauthors

In this section, we present an application of our computational trust model presented in Section 3.1.1 to the case of Wikipedia. The method allows a user to calculate trust score of all her partners she has collaborated with so far. The object of this task is to assign a contribution-based trust score to each partner, so we can assist users in assessing the trustworthiness of their partner.

As we discussed in the end of Section 3.1.2, in order to apply our computational trust model into real-world systems, we need to find a way to convert the behavior of users into numerical values. In Wikipedia, we propose to use the contribution values of users to represent their behaviors. We calculate the contribution value as the product of the size of a contribution by the quality of the article of this contribution.

Currently, the contribution of users to Wikipedia is measured quantitatively, i.e. the measurement counts only number of edits [Wikipedia, 2017c] made by users. However, several studies [B. Thomas Adler and Alfaro, 2007; B. Thomas Adler, Alfaro, et al., 2008] suggested that the measurement should take into account not only the number of edits but also the quality of edits. As the Wikipedia page “Edit Counts" claimed, “Edit counts do not necessarily reflect the value of a userś contributions to the Wikipedia project” [Wikipedia, 2017a].

It is difficult to measure quality of a single contribution itself, because usually the content of a modification along is not enough to analyze. We have to analyze the modification in its relations to the entire document. [Biancani, 2014] proposed to use crowd-sourcing system to measure the quality of each contribution. However, the work is just a proposal rather than a complete study.

To build a contribution measurement which takes the edit quality into account, we should rely on the quality of the article, because the quality of the article is made by all the contributions. We propose a novel method to calculate user contribution as follows.

Given an article A. We can scan through its entire history and determine the contribution

proportion of each contributors in term of number of characters, then we can normalize these proportions to the contribution score of each contributor.

We use Levenshtein distance [Levenshtein, 1966] to measure the contribution of each user. Levenshtein distance between two strings s and t is measured by total number of deletions, insertions and substitutions need to trans from s to t.

For the article A, we calculate the contribution proportion of each user to this article as the Algorithm 1.

Algorithm 1: Measuring contribution of users to an individual Wikipedia article.

Data: a Wikipedia article with complete history
Result: a dictionary, keys are users who contributed to the article and values are their contribution

```
// initialization
R := list of revision contents of the article;
U := list of users who contributed to each revision;
output := dictionary with keys as unique(U) and values are all 0;
N = |R|;
// measure the contribution
for i in 1:N do
    contribution := levenshtein (source = R[i-1], destination = R[i]);
    output[U[i]] += contribution;
end
output := normalize (output);
return output;
```

After calculating the contribution score in a single article, we can move on to calculate contribution score of a user through multiple articles. The main idea is to apply the computational trust model we presented in Section 3.1.1 for Wikipedians, but replace the sending proportion by the contributions of editors.

Consider a user Alice who wants to see how other partners contribute to her collaborative works. First of all, she should retrieve all the articles she was involved. After that, she could calculate the contribution score of each partner in each article. In the next step, she could multiply the contribution score of partner in each article by a weight number defined by the quality of the article. The main idea is, a contribution to a *FA* article should be counted more than a same contribution to a *Stub* article. In practice, we follow the previous studies [M. Hu et al., 2007; Suzuki, 2015] by assigning the score of 1 for the lowest quality class, then increase the score by 1 for each next higher quality class. For instance, for English Wikipedia, we assigned the score of 1 for Stub quality class, score of 2 for Start quality class, and so on. The final step is just to normalize the scores of all partners. We presented the pseudo-code in Algorithm 2. In this Algorithm, we presented also two baseline methods. The first baseline method considers all the contributions as the same quality level and use the sum of the contributions of a partner as the score. The second baseline method is similar with the first one but it takes into account the quality of Wikipedia articles.

In this section, we presented an approach of calculating trust score for a partner based on the assumption that we know the quality of all the articles that are shared between the user and the partner. In the next section we will present the experimental results in real-world Wikipedia datasets and discuss our approaches.

Algorithm 2: Measuring trust of Wikipedia editors.

Data: a Wikipedia editor u
Result: a dictionary, keys are partners who collaborated with this user and values are their trust score

```
// initialization
A := list of articles that the user involved;
Q := list of quality level of each article in A;
P := list of partners of u, in term they collaborate in at least one article in A;
output := dictionary with keys as unique(P) and values are all 0;
N = |A|;
M = |P|;
// measure the trust
for i in 1:N do
    for j in 1:M do
        contribution := contribution (article = A[i], user = P[j]);
        if use the first baseline method then
            output[P[j]] += contribution;
        end
        if use the second baseline method then
            output[P[j]] += contribution * quality_score (A[i]);
        end
        if use our trust model then
            output[P[j]] += trust_model (contribution * quality_score (A[i]));
        end
    end
end
output := normalize (output);
return output;
```

3.2.6 Experiments & Results

3.2.6.1 Datasets

We test our models on three Wikipedia datasets provided by Wikimedia Foundation: English, French and Russian Wikipedia [Wikimedia, 2017]. The distribution of each quality classes in each datasets are provided in Table 3.6. The datasets are balanced, i.e the number of articles that belong to different quality class are similar. This is a very important characteristic, because we can avoid many problems which occur only in imbalanced datasets [Branco et al., 2016].

	English	French	Russian
FA/ADQ/FA	4921	1500	1155
GA/BA/GA	4893	1500	1155
B/A/SA	4916	1500	1155
C/B/I	4908	1500	1155
Start/BD/II	4913	1500	1155
Stub/E/III	4917	1500	1155
IV			1155

Table 3.6: Distribution of articles by quality class in each datasets, ranked by order of quality class from the highest to the lowest.

3.2.6.2 Results

Measuring the quality of Wikipedia articles The results of different methods on measuring quality of Wikipedia articles are displayed in Table 3.7.

	English		French		Russian	
	accuracy	*AUC*	*accuracy*	*AUC*	*accuracy*	*AUC*
[Warncke-Wang, Ayukaev, et al., 2015]	59%	0.85	-	-	-	-
[Q. Dang and C. Ignat, 2016b]	63%	0.90	-	-	-	-
[Q. Dang and C. Ignat, 2016d]	55%	0.79	52%	0.75	50%	0.72
[Halfaker and Taraborelli, 2015]	62%	0.86	53%	0.82	56%	0.81
RNN-LSTM	68%	0.92	65%	0.84	63%	0.83

Table 3.7: Performance of different algorithms

We presented three different approaches in assessing the quality of Wikipedia articles. In this section we will discuss in details about the advantaged and disadvantages of each approach.

The first method [Q. Dang and C. Ignat, 2016b] was using the traditional machine learning approach: we define features by hand and apply several shallow machine learning algorithms on the defined features. The second method [Q. Dang and C. Ignat, 2016d] is mixed between shallow and deep learning techniques where we used *Doc2Vec* for automatic feature engineering then Deep Neural Networks on the output of *Doc2Vec*. The third method is an end-to-end deep learning solution where we feed raw Wikipedia contents into a RNN-LSTM model to predict directly the quality classes.

In the perspective of predicting performance, the method of using RNN-LSTM achieves the highest scores. On the other hand, the method of RNN-LSTM and *Doc2Vec*-DNN are language-neutral, means that they can be applied in any language, while the other method depends on

language. In fact, the method presented in [Q. Dang and C. Ignat, 2016b] is available only for English.

However, the method of RNN-LSTM has several disadvantages:

Computational time. The computing time of RNN-LSTM is much longer than [Q. Dang and C. Ignat, 2016b; Halfaker and Taraborelli, 2015]. In practice[24], it took several days for training one model and several hours for testing. By contrast the methods of [Q. Dang and C. Ignat, 2016b] or [Halfaker and Taraborelli, 2015] can return the results on the order of seconds.

Interpretation. Interpretation is another problem of machine learning in general, means that the machine learning model is not understandable from end user's point of view [M. T. Ribeiro et al., 2016]. While the prediction of [Q. Dang and C. Ignat, 2016b; Halfaker and Taraborelli, 2015; Warncke-Wang, Ayukaev, et al., 2015] is somewhat explainable in plain text, the results of LSTM model do not have this feature. For instance, the results of [Q. Dang and C. Ignat, 2016b; Halfaker and Taraborelli, 2015; Warncke-Wang, Ayukaev, et al., 2015] can be interpreted as suggestions like "with two more references you can improve the quality class of your article from Stub to Start". On the other hand, it is difficult to explain how can the RNN-LSTM model make a prediction.

To summarize, the RNN-LSTM method can be used in *offline* quality assessment, i.e. when we do not need the result be returned quickly. On the other hand, the method using random forest can be implemented as an *online* quality assessment, i.e. we can make a prediction right after a modification of a user.

Measuring trust of Wikipedia coauthors We collected a set of 400 Wikipedians and calculated trust score of users as described in Section 3.2.5. The baseline method to compare is to calculate the contribution of users regardless the quality of Wikipedia articles as being done currently in Wikipedia.

As we discussed in Section 1.4.2.2, a trust model is good if we can use the trust scores calculated by this model to predict future behavior of users. Therefore, we compare our proposed trust model with the baseline trust models to see whether our trust model can do better than the baseline one in predicting the future contribution of users.

We used the future contributions of partners as the the dependent variable, and used scores from three scoring methods we presented in Section 3.2.5 alternatively as the independent variables, then we fed them into linear regression analysis to see how each score relate to the future contribution. The results of the analysis are presented in Table 3.8. We see that, the scores computed by the first baseline method do not correlate with the future contributions. The scores computed by the second method correlate with the future contributions but with very low adjusted R-squared value. The scores computed by our trust model correlate with the dependent variable at much higher F-statistic and adjusted R^2 compare to the baseline methods.

Based on he experimental results, we claimed that our trust model can better predict the future behavior of users. Therefore, a user can rely on this trust model to see whether a partner will contribute in the future or not.

Our proposed trust model relies on the quality of sharing articles. If we can define quality models for other collaborative systems, we can easily extend the trust model to these systems.

[24] We used *Grid5000* [Balouek et al., 2012] to train and test the model on the cluster which is equipped with strong GPUs such as Titan X or K40. The least powerful cluster is equipped with 64GB of RAM.

	F-value	Adj.R-squared
Baseline_1	F(1,398) = 3.16	0.005
Baseline_2	F(1,398) = 4.06*	0.01
Our trust model	F(1,398) = 33.54***	0.1***

Table 3.8: Regression analysis on three scoring methods against future contribution in Wikipedia

3.3 Discussion

In this chapter, we discuss the second research question of the thesis: “How do we calculate trust score of users in a collaborative system?". We aim to design a trust model that calculate the trust score between a pair of a user and a partner in collaborative contexts. We assume that the user and the partner interacted with each other before.

We present a computational trust model to calculate a trust score of a user on a partner, based on an assumption that the user and the partner interacted before. The trust model takes into account only the behavior log of an user, i.e. the trust model relies only on activities that the user can observe. The model traces the behavior of partner and calculate the trust score of this partner.

We validate the model in two different contexts: trust game and Wikipedia. The two contexts are different in nature. Trust game is a lab-control experiment with a small number of participants compare to Wikipedia. Wikipedia is an online encyclopedia wherein people from all over the world can contribute their knowledge. We show that our trust model outperforms baseline models in both contexts. It is an indicator that human behaviors share similarities cross-domain. Therefore we can expect that our trust model can be applied in other collaborative systems.

In the next chapter, we study the relations between users who did not interact with each other.

Chapter 4

Predicting Trust and Distrust Relationship

Contents

In the previous chapters, we studied two research questions: "Should we introduce trust score to users?" and "How do we calculate trust scores between users that already interacted with each other?"

In this chapter, we will study the last main research question. Given two users that have not interacted with each other, how can we predict their trust or distrust relationship?

4.1 Introduction

In the previous chapter, we presented our computational trust model that can calculate trust score for any pair of interacted users. It is also important to predict the trust/distrust relations between users who have never interacted with each others. The reason is that modern collaboration networks are huge in term of number of nodes but very sparse [Leskovec et al., 2010b; J. Tang, Gao, et al., 2012]. It means that the number of established links is much smaller than the number of possible links.

In collaborative systems, the interactions and relationships between users form *implicit social networks* [Maniu et al., 2011; Rozenshtein et al., 2017], i.e. the relationships between users can be represented as a graph where nodes are users and edges are relations between them [J. Tang, Y. Chang, et al., 2016]. In collaborative systems like Wikipedia, users can express their positive/negative opinions on partners in voting for someone to be an administrator of particular Wikipedia pages [Burke and R. E. Kraut, 2008]. The positive or negative signs are considered as trust/distrust opinions from users to partners [DuBois et al., 2011; Bachi et al., 2012; Z. Wu et al., 2016]. These systems can be represented by a *signed directed network* [Song and Meyer, 2015], i.e. the edges in this network have direction and sign (positive/negative).

In this chapter, we address the third research question. The question is how can we predict a future relation from a user to a partner is trust or distrust.

In this problem, we are given a network. The links of the network are assigned with positive/negative labels. However, the sign of one link is missing. Using the information from the network, we need to infer the sign of the link. The problem is also called *link-sign prediction* in literature [Leskovec et al., 2010a].

A lot of existing link-sign prediction algorithms fall into graph-based algorithms [Jiang et al., 2016]. The algorithms take the graph topology as the only input data without knowing the details of the graph. These algorithms can be applied to any signed directed graph. In this chapter, we proposed a graph-based link-sign prediction algorithm. Therefore, along with Wikipedia which is one of two main case studies in this thesis, we take into consideration two other signed directed graph datasets which are widely used in literature to evaluate the link-sign prediction algorithms.

We will use the following datasets to evaluate our algorithm:

Wikipedia In Request for Admissionship (RfA) process [Burke and R. E. Kraut, 2008], Wikipedia users can vote *for* (positive) or vote *against* (negative) other users in the election to be an administrator of particular Wikipedia pages.

Epinions. Users on Epinions can express their opinions as *trust* (positive) or *distrust* (negative) to other users.

Slashdot. Users on Slashdot can tag other users as *friend* (positive) or *foe* (negative).

These datasets are used widely in literature to evaluate link-sign prediction algorithms [Leskovec et al., 2010a; Dubois et al., 2012; Bachi et al., 2012; Song and Meyer, 2015; You et al., 2016; J. Wang et al., 2017].

The link-sign prediction problem has some applications in collaborative systems. If we can predict precisely the future relations between users, we can:

- Assist user in making some decisions.
  - In collaborative editing systems, a user (Alice) can require the access right to a document which belongs to another user (Bob) but Bob does not know Alice yet. In this situation, we can assist Bob to make a decision of sharing or not the document with Alice.
  - In Wikipedia, we can recommend users to vote for or vote against a candidate in RfA process.
  - We can also suggest to a candidate which user will vote for her and which user will against her, so this candidate can adjust her strategy. For instance, she can try to increase the trust of the users who are not fond of her currently.

- Assist users in deciding to trust or not to trust a partner that she does not interact with.
- Link-sign prediction can enhance the quality of existing mechanisms which we discussed in Chapter 2 such as reputation score or review.
  - Suppose Alice wants to assess the trust level of Bob in a system that is using a reputation scheme. She realizes that Bob is rated by 100 users and the average score of Bob is 4.2. However, if we can predict precisely the unknown relations, we can discover that Alice only trusts 60 users in 100 users who rated Bob. Therefore Alice only care about the rating score from these 60 users. In this case, the score of Bob becomes 2.5. Hence, Alice has a better view about Bob from people she trust rather than everyone.

4.2 Background Knowledge

In this section, we briefly describe some important properties of graphs in general and social graphs in particular for the task of predicting trust/distrust relationship.

4.2.1 Network properties

There are several ways to classify networks [Takemoto and Oosawa, 2012].

On the difference of links, there are undirected and directed networks. The links might be unsigned or signed. In this chapter, we only focus on signed directed network data.

On the topology of the network, we have two network classes: classical networks, i.e. a network is artificially established using some predefined rules, and scale-free networks that are real-world social network data [Takemoto and Oosawa, 2012].

There are several interesting properties of real-world social network data:

Connected Components Connected component (or *weakly* connected component for directed graphs) is defined as a set of nodes in a graph that there exists a path between any two nodes in the set.

Studies suggested that giant connected components form in real-world social networks [McGlohon et al., 2011]. For instance, [Ugander et al., 2011] claimed that the largest connected component of Facebook contains 99.1% of Facebook users. A same phenomenon is observed in other social networks [Leskovec et al., 2010a; J. Tang, Gao, et al., 2012].

We can interpret this property as in a network, everyone relate to everyone. We can predict the sign of almost any link in the network based on the information from the rest of the network.

Small-world phenomenon A classical theory suggested that all people in the world are separated by at most of six degree of separation [Guare, 1990]. In the Internet era, the degree of separation in online social networks decreases dramatically to 4 [Backstrom et al., 2012] and the most recent value is only 3.57 for Facebook users [Edunov et al., 2016].

Due to this property, we will focus on local neighborhood of a link for the sign prediction task rather than focus on the whole network.

Heavy-tailed distribution Distributions of node degree in social networks usually follow power-law distribution [Takemoto and Oosawa, 2012; X. Zheng et al., 2015; J. Tang,

Gao, et al., 2012; Leskovec et al., 2010a; Perozzi et al., 2014] as follows [McGlohon et al., 2011]:

$$y(x) = Ax^{-\gamma} \tag{4.1}$$

wherein A and γ are two positive constants.

Power law distribution means that, while there is a small number of popular nodes, i.e. these nodes have a lot of connections, most of nodes have few connections. The power law distribution is observed in almost every existing networks. In fact three datasets we used in this chapter (Wikipedia, Epinions, Slashdot) follow the power-law distribution [Maniu et al., 2011; Dong et al., 2012]. It suggests that if we build a graph-based link-sign prediction algorithm, we can apply this algorithm on different social graphs if they follow the same distribution.

4.2.2 Graph sampling

We define a graph $G =< V, E >$ as a set of vertices V and edges E. We denote W as a weight matrix for edges. W has $|V|^2$ cells. Each cell w_{ij} of the matrix W could contain one of three values: 1 means that there is a positive link from node i to node j, 0 means there is no link and -1 means there is a negative link.

There are two extreme sampling methods which are popular for graph data [Cormen et al., 2009, Chapter 22]:

- Breath-first Sampling (BFS).
- Depth-first Sampling (DFS).

Both of these extreme sampling will cover the entire graph.

However, as studies [Leskovec et al., 2010a; Song and Meyer, 2015; X. Zheng et al., 2015] suggested, the information of a node is mostly influenced by neighbor nodes rather than the entire network. For instance, a decision made by a user in Chile probably has no influence on a user in South Africa, given that there are no direct relationship between them.

[Manning et al., 2008, Chapter 21] defined a random walk on nodes as a series of nodes $v_1, v_2, ..v_n$ wherein v_i and v_{i+1} are immediate neighbours for $1 \leq i \leq n-1$. n is the number of steps of the walk.

We define a random walk formally as follow [Grover and Leskovec, 2016]. Given a starting node u and a length n of the walk. We denote c_i as the i^{th} node in the walk, with $c_0 = u$.

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \beta * \frac{\pi_{v,x}}{Z} \text{ if } (v,v) \in E \\ (1-\beta) \text{ otherwise} \end{cases} \tag{4.2}$$

wherein $\pi_{v,x}$ is the transition probability between two nodes v and x, Z is the normalizing constant, and β is a random jump constant [Page et al., 1999].

4.2.3 Link analysis tasks

[J. Tang, Y. Chang, et al., 2016] distinguished different link analysis problems. We visualize them in Figure 4.1. In this Figure, subfigure (a) represents the link prediction problem [Liben-Nowell and Kleinberg, 2007], subfigure (b) represents the link-sign prediction problem [Leskovec

et al., 2010a] and subfigure (c) represents the negative link prediction [J. Tang, S. Chang, et al., 2015].

In this thesis, we only focus on link-sign prediction problem, because this problem fits to the problem of trust/distrust prediction.

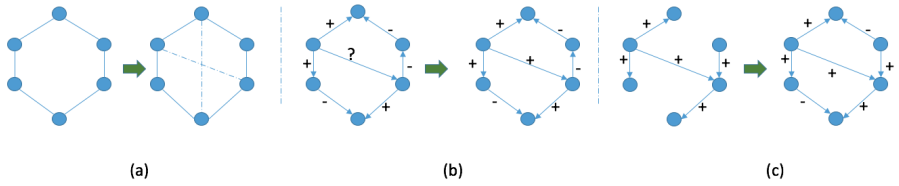


Figure 4.1: Different link analysis tasks [J. Tang, S. Chang, et al., 2015].

4.3 Related Work

Several studies have been presented for the problem of trust/distrust prediction, or link-sign prediction in signed directed graphs [Song and Meyer, 2015]. Here we only present the research works which belong to graph-based approaches [Jiang et al., 2016] that do not require any external information, such as personal information or users' historical trading information. The input information of these algorithms are uniquely graph topology as visualized in Fig. 4.1.

Graph-based link analysis algorithms are proposed because of two reasons:

- These algorithms can be applied in an arbitrary graph without knowing details information about graph.
- These algorithms used a minimal personal information. In fact, some link signs are easily inferred by using personal information [J. Tang, Gao, et al., 2012]. For instance if we know two users are a couple it is easy to infer with a high confident level that they will maintain positive links between them. However, due to the raising privacy concerns on the Internet the usage of personal information should be avoided.

[Guha et al., 2004] presented one of the first prediction by using a trust and distrust propagation framework. The authors defined four atomic propagating operators which can be described in natural language as "if A trusts B and B trusts C so A trusts C", "if A trusts C and D and B trusts C so B trusts D", "if A trusts B and C trusts B so C trusts A" and "if A and B trust D and C trusts A so C trusts B". The prediction is executed by recursively applying these atomic operators on users' relations matrix. In theory the propagation could be performed until all missing links are predicted. However, longer propagation tracks lead to lower confidence of the prediction results.

Several algorithms rely on two social psychology rules: *structural balance theory* and *social status theory*. In short, *structural balance theory* states that, a triad which represents relations between three users tends to be balanced, i.e. it has an odd number of positive signs regardless the direction, as we visualize in Figure 4.2. *Social status theory* claims that, if there is a positive edge from A to B, then A considers herself having a *lower* social status than B, and if there is a negative edge from A to B, then that A consider herself having a *higher* social status than B. In this thesis, we will use the notation $A \xrightarrow{+} B$ to state that there is a positive link from A to B, $A \xrightarrow{-} B$ to state that there is a negative link from A to B, and $A > B$ to state that A has a

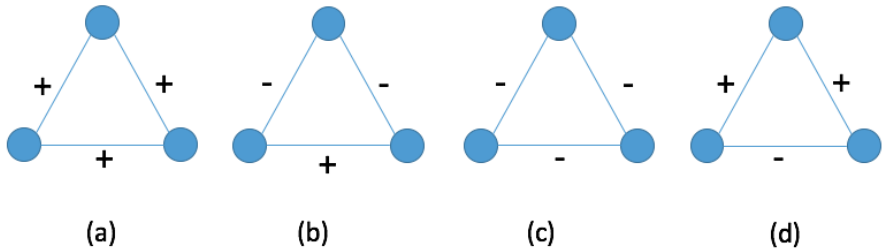


Figure 4.2: Visualization of structural balance theory [Leskovec et al., 2010a]. According to *structural balance theory* [Leskovec et al., 2010a], triads *(a)* and *(b)* are balanced, while *(c)* and *(d)* are not. According to *weak balance theory* [Hsieh et al., 2012; Leskovec et al., 2010a], triads *(a), (b)* and *(c)* are balanced, while *(d)* is not. Structural balance theory does not take the direction of edges into consideration.

higher social status than B. We use $A \rightarrow B$ to state that there is a link from A to B regardless the sign.

Using the above notations, we could express *social status theory* as, if $A \overset{+}{\rightarrow} B$, then $A < B$, and if $A \overset{-}{\rightarrow} B$, then $A > B$. If everyone agreed on a common social status, we could make a prediction as, if $A > B$ then $A \overset{-}{\rightarrow} B$ and $B \overset{+}{\rightarrow} A$. We visualize social status theory in Figure 4.3.

Based on these two theories, [Leskovec et al., 2010a] trained a logistic regression on a set of seven degree features calculated from triads of the signed directed social network graphs. [Chiang et al., 2011] extended the research work of [Leskovec et al., 2010a] by using longer cycles such as quadrilaterals or pentagons. [Hsieh et al., 2012] presented low-rank matrix approximation with *weak balance theory*, which extended the *structural balance theory* by considering a triad with all three negative edges as a balance triad. [You et al., 2016] combined the two social theories with users trustworthiness and predict how likely a user will trust other users. [Zhou et al., 2014] presented a technique called PLSP which uses parallel programming to speed up the training speed of classifier based on social psychology theories. The approach achieves good performance, but requires global information and also other external information such as reviews of users on other users.

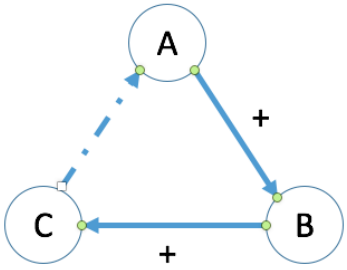


Figure 4.3: Visualization of social status theory [Leskovec et al., 2010a]. The sign of the dash line from C to A is inferred by their social status. Because $A \overset{+}{\rightarrow} B$ and $B \overset{+}{\rightarrow} C$, therefore we have $B > A$ and $C > B$, so $C > A$, hence social status theory predicts that the sign of line from C to A is negative.

[DuBois et al., 2011] developed an algorithm by combining path-probability trust inference algorithm with spring-embedding technique for trust / distrust prediction. The proposed algorithm requires a global view on the entire network. Although the algorithm performs well in density networks, i.e. where vertices of networks form triangles, its performance decreases dramatically in sparse networks.

[Song and Meyer, 2015] argued that, (i) even structural balance theory and social status theory contributed played an important role in existing research studies on link sign prediction, these theories are not very suitable in large-scale and extreme sparse network, and (ii) a fully observed network is not always available in practice, and developed a Bayesian node features based on partially observed networks. Finally the authors used a logistic regression classifier for link sign prediction problem. However, the performance of the approach is not very good compare to other recent studies.

Deep learning is gradually adapted in recommendation on graphs. On using deep learning for feature selection of graphs before applying other "shallow" machine learning algorithms, [F. Liu et al., 2013] used Deep Belief Network (DBN) on node degree feature sets, while [S. Deng et al., 2016] applied deep autoencoder for feature selection in social recommendations, and [Xiaoyi Li et al., 2014] used Restricted Boltzmann Machine (RBM) for feature selection. All the approaches presented in [F. Liu et al., 2013; Xiaoyi Li et al., 2014; S. Deng et al., 2016] required re-training when the network changes.

[Covington et al., 2016] presented the usage of feed-forward neural networks for Youtube recommendation. The solution is being used by Google. However, the solution requires to access personal information.

Inspired by recommender systems, [You et al., 2016] considered trust / distrust declarations from users to other users as recommendations, and applied matrix factorization, which is well-known and has been used for a long time in recommender systems, for link sign prediction problem.

Several research studies focus on a more general problem that is network embedding problem. The network embedding algorithms aim to represent a network by a low-dimensional matrix. We can apply a conventional machine learning algorithm on the output matrix for different predictive tasks including link-sign prediction [Grover and Leskovec, 2016]. [S. Wang et al., 2017] presented an algorithm called *SiNE* that uses multi-layer feed-forward neural network to learn the representations of a signed graph. The authors designed the objective function of the neural network based on structural balance theory, i.e. the neural network tries to approximate the similarities between users based on the structural balance theory. [Yuan et al., 2017] presented an algorithm called *SNE* that uses the skip-gram model to learn the similarities between nodes in a signed graph.

Many presented approaches require a fully observed network which consumes a lot of computational power while the long running time is ignored [Zhou et al., 2014]. Furthermore, none of existing approaches consider dynamic graphs. In other words, the existing approaches took a snapshot at a particular point of time of a network then do the analysis. If the graph changes, the prediction needs to be performed from the beginning.

Real-world social networks change frequently. The reason is there are a huge number of active users in popular social networks and their activities change the network topology. For instance, Facebook reported that the company has 1.23 billion daily active users on the last day of 2016[25]. The network topology of real-world social networks change every second, or even faster. It leads to several critical issues:

[25] http://newsroom.fb.com/company-info/ accessed on 14-Feb-2017.

- It is very difficult to capture a full snapshot of a social network [Morstatter et al., 2013]. In fact, the most popular method to capture a snapshot of a social network is to start from several seed nodes then recursively collect the neighbors of these seed nodes [Leskovec et al., 2010b; Grover and Leskovec, 2016] by the graph sampling techniques we discussed above. However, the network itself will change during the graph sampling process, so the graph at the beginning and at the end of the process are different. In order to capture a precise snapshot of a network we have to *freeze* the network during the sampling process which is definitely impossible.

- The existing approaches do not utilize the time information. Studies [J. Tang, Gao, et al., 2012; Ostrom, 2014] claimed that time plays an important role in forming social relationship. For instance, a friendship which was formed 20 years ago should has a different influence from another friendship which was formed two hours before. However, existing approaches treat these two relations as the same.

- Furthermore, by eliminating time information, we might bias the model by learning wrong information. The reason is, when we remove the time information, we implicitly assume that the link with missing sign is established *after* other links with known signs. In training phase, the learning algorithm will learn the topology corresponding with a sign but that topology might be created after the link, so the topology does not reflect the observation of a user when she establish the link.

  – Consider an example graph with five nodes: Alice, Bob, Carol, Dave and Evie. In day 1, Alice established a positive link to Carol, i.e. Alice trusts Carol. In day 2, Carol established a positive link to Bob. In day 3, Alice wanted to form a link to Bob. She looked to her topology and realized that one her trusted friends, Carol, trusts Bob, so Alice formed a positive link to Bob. In day 4, both Dave and Evie formed negative links to Bob. If we feed the topology after day 4 to a learning algorithm, the algorithm will learn that Alice formed a positive link to Bob when a majority of her friends formed negative links to Bob, despite the fact that when Alice made a link these negative links did not exist yet.

- The existing approaches do not utilize trained data. Training in general is an expensive task in large-scale machine learning. On the other hand, new arriving data does not change the previous one, which means that a new link in the network does not change the status of established links. Therefore, it is better to perform incremental learning, i.e. that the model only needs to learn new data when it arrives rather than learn everything from beginning. We note that, several incremental training solutions for logistic regression which are used by previous studies [Leskovec et al., 2010a] are available. However, the concerns are not only that the previous studies do not utilize them but also they require performing feature engineering process from beginning when the network changes. For instance, when the network changes, the algorithm of [Leskovec et al., 2010a] requires scanning the network again to assign new features to nodes.

To solve the above issues, we propose a novel incremental learning approach using random walk and stateful LSTM for the problem of trust/distrust prediction.

Similar to our approach, [R. Agrawal, Alfaro, and Polychronopoulos, 2016] learned a graph neighborhood by using LSTM. However, the authors used tree-based sampling approach while we used random walk and they do not consider the time information. As we discussed in previous section, tree-based sampling is an extreme sampling method that will travel the graph entirely.

4.4 Our Approach

In this section we present our approach for link-sign prediction problem.

The approach can be divided in two steps. In the first step, we perform a sampling process then measure the distance between nodes. In the second step, we feed the action list of a node as a time-series data into RNN-LSTM for prediction.

4.4.1 Node distance by random walk & Doc2Vec

The task of link-sign prediction is, given a link from node A to node B, we need to predict the sign of the link. As discussed above, the first step is to measure the distance between A and B.

There are two main approaches to measure the node distance in graphs which are *global-based* measurement and *local-based* measurement [Even, 2011]. Global-based measurement means that the full observation of the graph is available and local-based measurement means that only local topology around current interest nodes are available.

We chose local-based measurement in this thesis because of two reasons:

- As we discussed above, the full observation of real-world networks is not available.
- As studies [DuBois et al., 2011; Song and Meyer, 2015; Cygan et al., 2015] suggested, the decision of a user is influenced by their directed friends, and the influence becomes weaker quickly while the distance between two users increases. It is not necessary to acquire global information of the network to predict sign of only one link.

The distance measurement task can be further divided in two smaller tasks: graph sampling and vector mapping.

4.4.1.1 Sampling by Random Walk

Random Walk is used widely in graph sampling [Leskovec and Faloutsos, 2006; Vishwanathan et al., 2010; B. F. Ribeiro and Towsley, 2010; Perozzi et al., 2014; R. Li et al., 2015; Grover and Leskovec, 2016; Kipf and Welling, 2017].

In order to measure the distance between nodes, we perform random walk for each node. The random walk we used is similar with [Grover and Leskovec, 2016], i.e. we follow the edges regardless the direction with the transitional probability. For instance, we can perform a step from node A to node B even there is only link from B to A.

Let's consider an example of a walk as visualized in Figure 4.4. Suppose that the walk has just moved from node A to B and now the walk is staying in node B. Now we form the unnormalized the transitional probability, i.e. the probability of following node, as follows:

$$\begin{aligned} \alpha(A) &= \frac{1}{p} \\ \alpha(x|\text{there is a link between } A \text{ and } x) &= 1 \\ \alpha(x|\text{there is no link between } A \text{ and } x) &= \frac{1}{q} \end{aligned}$$

There is an unnormalized probability of $\frac{1}{p}$ for the walk to immediately come back to the previous node (node A). Similarly, the probability of $\frac{1}{q}$ is the probability that the walk further explores the part of the network which has not been explored before. Different from Node2Vec [Grover and Leskovec, 2016], we keep the sign of visited links through the walk.

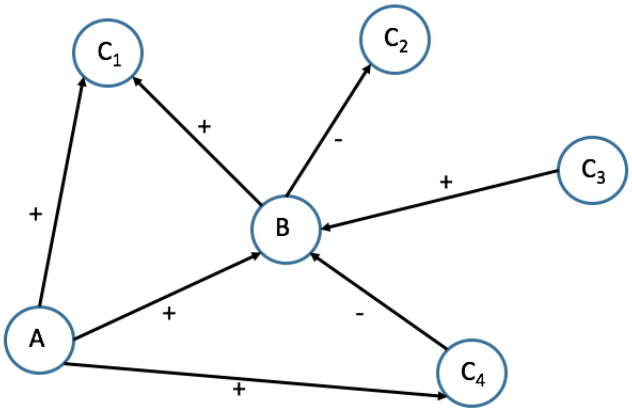


Figure 4.4: Graph sampling by random walk.

4.4.1.2 Vector mapping

Now for each node, we have a list of nodes as the result of random walk. Recent studies in graph embedding utilize the techniques in natural language processing [Perozzi et al., 2014; Ristoski and Paulheim, 2016; Grover and Leskovec, 2016].

Existing studies usually rely on Word2Vec [Mikolov et al., 2013] for mapping a series of node to a vector. However, because the links in our case are signed links, we used Doc2Vec [Le and Mikolov, 2014] instead of Word2Vec for the task.

The vector mapping algorithm is described in Algorithm 3.

Algorithm 3: Graph Vectorizing

Data: a signed directed graph $G =< V, E >$
Result: a list of vector, each vector represents a node in the graph.
`// initialization`
walks := an empty vector;
output := an empty vector;
$N = |V|$;
`// random walk`
for *i in 1:N* **do**
    w := RandomWalk (V[i]);
    walks.append (w);
end
`// vectorize`
output := Doc2Vec (walks);
return *output*;

4.4.2 Recurrent Neural Networks for Relationship Prediction

After using Doc2Vec for transforming a node series to a vector, the final tasks to predict the sign of the link from node A to node B are:

- Initialize an empty vector.

- For each link established from A to x, calculate the distance from A to x.
- For each distance value calculated, if the sign is negative, put the negative value of the distance value to the vector. Otherwise, put the distance value to the vector.
- Feed the vector to the RNN.

We display the pseudo-code of using RNN-LSTM in Algorithm 4. In this Algorithm, we used the argument $length = 1$ for RNN to predict the sign of the next link. The main idea of the algorithm is that we consider a link established from A to B as a *step* from A. Given the list of previous steps of A over time, we feed the data into a RNN model to predict the next step made by A.

Algorithm 4: Sign Prediction

Data: output of the Graph Vectorizing task $graph\_vectors$.
Data: the graph $G =< V, E >$
Data: two nodes A and B whose the link has unknown sign.
Result: predicting sign of the link $A \rightarrow B$.
`// initialization`
$index_A := V.index(A)$;
$index_B := V.index(B)$;
$distance\_vector := anemptyvector$;
$K := length(graph\_vectors[index_A])$
`// distance calculation`
for *i in 1:K* **do**
  $d := cosine\_distance(graph\_vectors[index_A][i])$;
  $distance\_vector.append(d)$;
end
$sort(distance\_vector, key = established\_time)$;
`// sign prediction`
$rnn := RNN\_LSTM(distance\_vector)$;
$raw\_predict := rnn.predict(length = 1)$;
sign := ifelse $(raw\_predict > 0,1,-1)$;
return *sign*;

4.5 Experimental Results

4.5.1 Datasets

The RNN-LSTM algorithm for link-sign prediction is validated against three popular signed directed real-world datasets: Epinions, Slashdot and Wikipedia[26]. The datasets are collected by [Leskovec et al., 2010b].

- Epinions is a product review website. Users of Epinions can explicitly state their trust (positive) or distrust (negative) opinions on other users.

[26]The datasets can be obtained at http://snap.stanford.edu

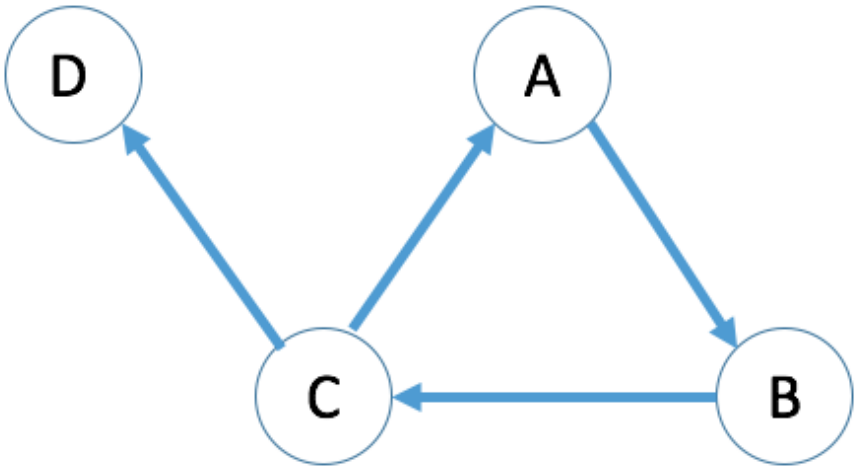


Figure 4.5: Visualization of connected component. The set of three vertices A, B and C is a strongly connected component, while the set of all four vertices is a weakly connected component (WCC).

- Slashdot is a website focusing on technological news. Users on Slashdot can declare other users as friend (positive) or foe (negative).
- Wikipedia dataset contains voting results for Wikipedia Request for Adminship (RfA) process. A user can vote for (positive) or against (negative) other users to become administrators of Wikipedia pages.

Several basic statistics of three datasets are displayed in Table 4.1.

The 1^{st} and 2^{nd} row of the table displayed number of vertices and edges in each dataset. The 3^{rd} row showed the fraction of existing edges over the total number of possible edges, i.e. the number of edges if the network is fully connected. We could see that all graphs are extremely sparse.

The $4th$ and 5^{th} row displayed the distribution of positive and negative edges on each dataset. Most of edges are positive, therefore a predicting model need to provide a prediction with accuracy higher than the percentage of positive edges. For instance, a predicting model which provides a prediction with accuracy of 84% on Epinions dataset is nonsense, because a naive approach which predicts every output as positive will achieve the accuracy of 85%.

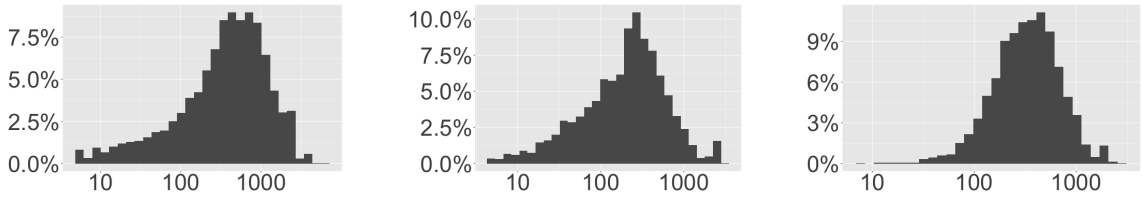
The value “largest WCC" presented in the 6^{th} row of Table 4.1 showed how many percent of total edges belong to the largest weakly connected component in each dataset. We visualize WCC in Figure 4.5. We claim that these graphs are weakly connected. The finding is consistent with other OSN platforms. For instance, 99.91% of Facebook users are connected [Ugander et al., 2011].

The 7^{th} row of Table 4.1 presents the average size of primary neighborhood sets of all edges in each datasets. The size of primary neighborhood set of node A is the number of nodes that have direct connection with A regardless the direction. The details distribution of primary neighborhood set size is displayed in Figure 4.6. The histograms show that the distributions of primary neighborhood set size are similar between datasets.

The 8^{th} row “fraction of triads" presents the fraction of number of existing triads over total number of possible triads in each dataset. We could see that these fractions are extremely small,

	Epinions	Slashdot	Wikipedia
# of nodes	119 217	82 140	7 118
# of edges	841 200	549 202	103 747
fraction of edges	$6e^{-5}$	$8e^{-5}$	$2e^{-3}$
+ edges (%)	85.0	77.4	78.8
− edges (%)	15.0	22.6	21.2
largest WCC (%)	99.1	100	100
average # of directed connection	590	327	418
# of triads	13 375 407	1 508 105	790 532
fraction of triads	$1.35e^{-10}$	$5.46e^{-11}$	$4.25e^{-9}$

Table 4.1: Basic statistics of datasets. WCC stands for *weakly connected component*.



(a) Epinions (b) Slashdot (c) Wikipedia

Figure 4.6: Distribution of size of primary neighborhood sets in three datasets (log scale)

i.e. triads are not popular in all three datasets. Therefore, the algorithms rely on sociology rules [Leskovec et al., 2010a; Hsieh et al., 2012] might not perform well on these datasets.

4.5.2 Experiments on Static Graphs

In this section, we perform link-sign prediction in static graphs, i.e. the graphs where all nodes and links are available, and there is no removal or addition of nodes or links. We follow the leave-one-out validation setting of [Leskovec et al., 2010a], i.e. we alternatively remove the sign of one link and try to predict this sign. Finally we compare the prediction with the ground truth.

All three datasets are highly imbalanced, therefore *accuracy* score is not the most suitable metric to evaluate the algorithms. However, due to the fact that most existing studies used the

Size	Epinions	Slashdot	Wikipedia
100	16.15%	26.95%	6.92%
200	27.46%	46.24%	25.44%
300	37.82%	64.37%	44.02%
400	47.53%	75.06%	60.93%
500	55.54%	82.20%	70.87%
1000	81.25%	95.01%	93.44%

Table 4.2: Cumulative distribution of primary neighborhood set size. For instance, on Epinions dataset, there are 16.15% of edges that have the size of primary neighborhood set smaller or equal 100.

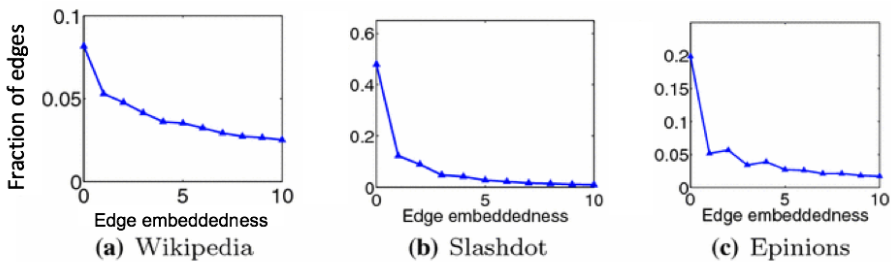


Figure 4.7: The distribution of edge embeddedness in three datasets [Song and Meyer, 2015]

accuracy score to report the performance of their algorithms, we keep using this metric for the comparison purpose. We also report F1-score but only for further references.

	Epinions	Slashdot	Wikipedia
Degree features[Leskovec et al., 2010a]	90.39	83.76	83.58
Triad features [Leskovec et al., 2010b]	90.42	80.42	82.46
Degree + triad features [Leskovec et al., 2010a; Leskovec et al., 2010b]	92.25	84.91	84.87
Longer cycles features [Chiang et al., 2011]	90.64	83.83	84.04
Spring-based inference [DuBois et al., 2011]	89	82	81
Low-rank modeling [Hsieh et al., 2012]	92.48	84.57	84.93
Weighted MF-LiSP [P. Agrawal et al., 2013]	89.0	80.2	80.0
PLSP [Zhou et al., 2014]	96.2	89.6	89.1
Bayesian-based model [Song and Meyer, 2015]	93.61	85.24	87.28
ESS [G.-N. Wang et al., 2015]	95.0	88.08	-
PMF [You et al., 2016]	94.06	91.28	-
RNN-LSTM	**96.31**	**91.66**	**89.76**

Table 4.3: Link Sign Prediction Accuracies (%). The best accuracies are highlighted in bold. The values are extracted from corresponding papers. The metric of Wikipedia prediction by ESS and PMF are missing because the authors do not present the performance of these algorithms on Wikipedia dataset.

We presented the accuracy scores on three datasets in comparison with state-of-the-art solutions in Table 4.3. Our algorithm outperforms other state-of-the-art algorithms in term of *accuracy* score in all three datasets.

We presented the F1-score of the RNN-LSTM approach with other algorithms in Table 4.4. However, we note that the F1-score for these other baseline algorithms are based on our own implementation of these algorithms, therefore they might not reflect their true performance.

4.5.3 Experiments on Dynamic Graphs

In this section, we consider the link-sign prediction problem in dynamic graphs, i.e. the graphs when links are added over time. Because there is no existing study link-sign prediction in dynamic graphs, we reimplemented two algorithms presented by [Guha et al., 2004] and [Leskovec et al., 2010a] as baseline algorithms.

We first established the network by adding links one by one. When the number of links reach to $1,000$, we start the prediction. We fed the next link into each algorithm, namely the

	Epinions	Slashdot	Wikipedia
Trust propagation [Guha et al., 2004]	0.892	0.885	0.882
Degree features [Leskovec et al., 2010a]	0.889	0.893	0.887
RNN-LSTM	0.911	0.905	0.896

Table 4.4: F1-score of different algorithms on static graphs. The F1-score of two baseline algorithms are based on our own implementation.

algorithm of [Guha et al., 2004], the algorithm of [Leskovec et al., 2010a] and ours. After all three algorithms made the prediction, we added this new link into the training set and fed the next link. All the experiments are executed on Grid5000 [Balouek et al., 2012] server. In order to make a fair comparison, we executed our algorithm on CPU mode.

As we described above, the approach presented by [Guha et al., 2004] is a simple rule-based approach. The approach is implemented as matrix operations that requires constant running time regardless of input size. The approach presented by [Leskovec et al., 2010a] is a logistic regression based approach, so the running time should increase linearly with the input size [Minka, 2003]. Our algorithm relies on stateful RNN-LSTM, so we could expect a long running time in the beginning and stable running time in next predictions.

We display the running time of three algorithms in Figure 4.8. The running time of the trust propagation algorithm [Guha et al., 2004] is constant regardless of the size of the dataset, while the running time of the logistic regression based on sociology rules [Leskovec et al., 2010a] increases almost linear with the graph size. The observations confirmed our theoretical predictions about the running time of each algorithm.

Similarly, we display the accuracy score of three algorithms on dynamic graphs in Figure 4.9. Again, we could see that the performance of the trust propagation [Guha et al., 2004] does not depend much on the size of dataset, while the logistic regression based approach [Leskovec et al., 2010a] performs better when there are more data available. The RNN-LSTM also achieves higher score with more data but the influence of new data is less than the method of [Leskovec et al., 2010a]. Furthermore, the accuracy scores we achieved with our implementation are similar with the scores reported in [Guha et al., 2004] and [Leskovec et al., 2010a]. It confirmed that the performance of our implementation is not far from the original ones.

4.6 Discussion

We presented an approach of combining Random Walk, Doc2Vec and RNN-LSTM for link-sign prediction in signed directed networks. We recall that, while the proposed algorithm can be applied to an arbitrary network, the original objective is to predict the trust/distrust relations between users in collaborative systems as we discussed in Section 1.2.3. We used the Wikipedia dataset as the main validation dataset, but we included Epinions and Slashdot datasets for external validation. The fact that our algorithm performs well in all three datasets despite the fact that they are collected from different websites, allows us to expect that our algorithm can perform well in other systems.

We showed that our algorithm outperforms state-of-the-art algorithms in link-sign prediction for static graphs. We showed that, in dynamic graphs where nodes and links are added, our algorithm outperforms two well-known link-sign prediction algorithms of [Guha et al., 2004] and [Leskovec et al., 2010a]. In fact, the running time of our algorithm is very high compared to other two algorithms in beginning, but it require almost constant running time when new links

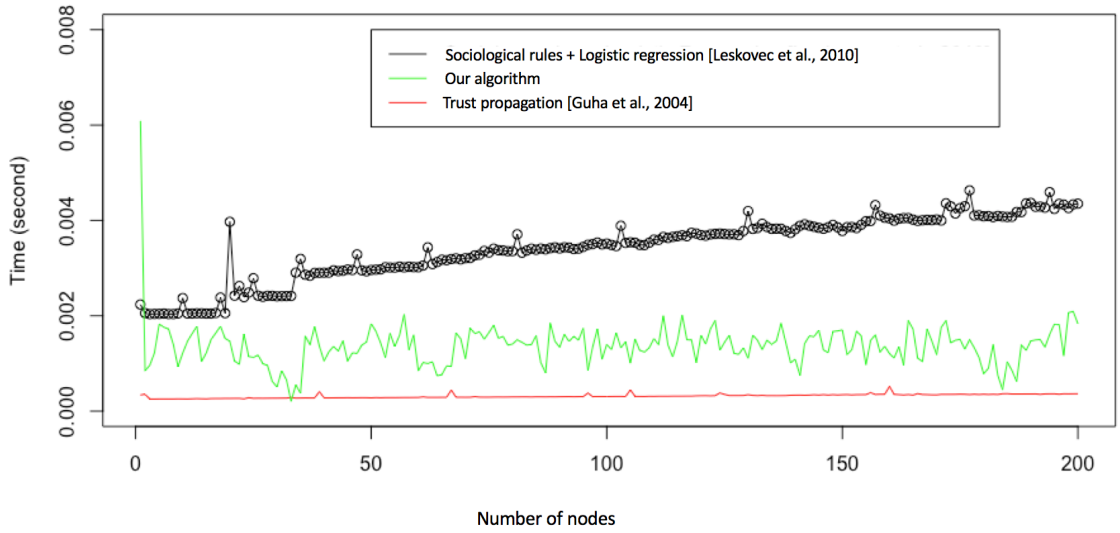


Figure 4.8: Running time of different algorithms on dynamic graphs.

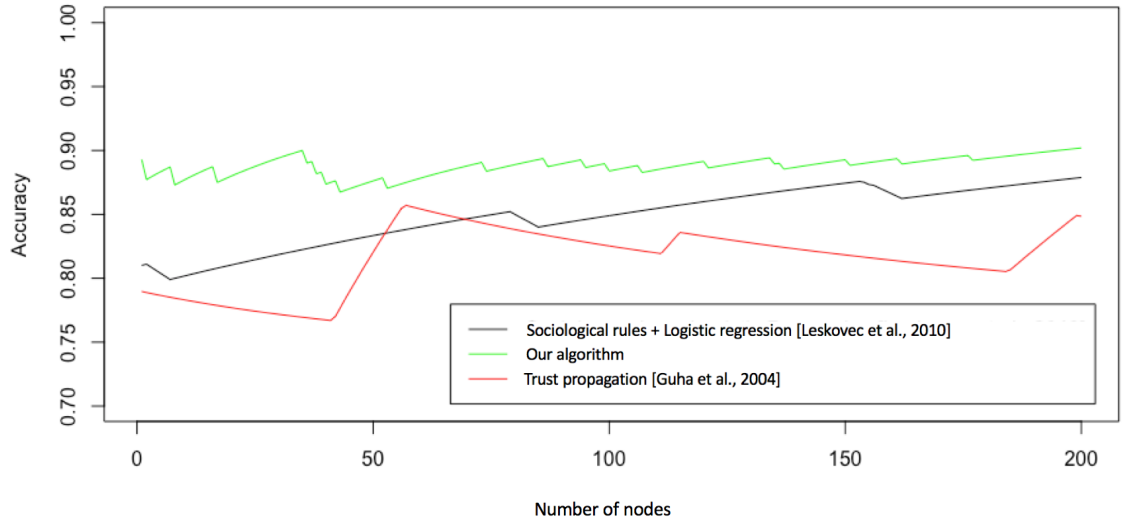


Figure 4.9: Accuracies on dynamic graphs.

are introduced. It makes our algorithm suitable for deploying in real networks. Nonetheless, our algorithm shares a same limitation as we discussed in Section **??**: it is more difficult to explain the algorithm to users than some rule-based algorithms such as [Guha et al., 2004] or sociological-rules based algorithms such as [Leskovec et al., 2010a].

In fact, when a user expressed their trust/distrust opinion on other partner in Wikipedia RfA process, the user might know the partner already. However, this information is not available in the dataset, so our algorithm treats the user and the partner as they did not interact in the past. Therefore, our algorithm can be used to predict the future trust/distrust opinions between users who do not know each other.

Furthermore, our algorithm does not reveal personal information of users. Studies showed that users hesitate in expressing explicitly their opinions, particularly negative ones, on other

users [Massa and Avesani, 2007]. Our algorithm used an anonymous graph as an input to avoid the leak of personal opinions.

In Wikipedia, users can explicitly express their trust/distrust opinions on other users. These explicit relationship might not be available in other collaborative systems such as Google Docs. However, in these systems, we can consider a different kind of trust expression, such as if Alice grants an access to a document for Bob, or if Alice has collaborated with Bob for a long time. These relationship could be explored in future research.

In this chapter, we presented our algorithm to predict future relationship between users who did not interact with each others. It finishes the thesis. In the next chapter, we conclude the thesis and present some future research works.

Chapter 5

Conclusions

Contents

In this thesis we presented our studies on trust assessment in large-scale collaborative contexts from different views. We argued that, in large-scale collaborations, it is difficult for a typical user to assess the trust level of every partner she have. We argued that the conventional trust assessment techniques such as avatar or nick-name are either vulnerable or not scalable. The core idea of the thesis is that we can assist users in collaboration by introducing personal trust score that is unique for a pair of a user and a partner.

In order to bring trust into computer-based collaboration, we need a computational trust model.

We addresses three main research questions in the thesis:

1. Should we introduce trust score to users?
2. How to calculate trust score for a pair of users who interacted?
3. How to predict trust relations between users that have not interacted with each others?

5.1 Outcomes

In this section, we summarize the outcomes of the thesis for three research questions we presented in Section 1.2.

5.1.1 Influence of trust score on user behavior

Studies proposed different computational trust models in literature. However, to the best of our knowledge we are not aware of a research work that study the effect of trust score on user behavior. Particularly we study the effect of trust score on cooperative behavior.

We deployed trust game [Berg et al., 1995] which is widely used to understand human trust [Chakravarty et al., 2011; Dinesen and Bekkers, 2015] to study the influence of trust score in user collaboration. We organized experiments with participants recruited through public announcement at our institute. We collected user behavior log and analyzed the behavior statistically. :

We demonstrated that introducing either trust score or user ID significantly improve user collaboration with no additive effect. In comparison with simple repeated trust game when we show no information to users, in other games when we show either trust score or user ID to users, the sending proportion between users are higher significantly. We do not observe any additive effect, i.e. in case we show both trust score and user ID to users, the sending proportion is similar with the case we show only one of the two information.

We showed that users follow the suggestion of trust score. The sending proportion of users to partners are correlate with the trust scores of partners, i.e. users tend to send higher amount to partners with higher trust scores.

Trust score is better than reputation score in predicting future behavior of users. We used either trust score or reputation score as independent variables to predict future sending proportion of users. The experiments showed that trust score is better than reputation score in predicting future behavior.

We verified the findings by using a comprehensive statistical analysis.

As we discussed in Section 2.6, studies suggested that the findings of trust game experiment can be extended in real-world systems, i.e. it could be valuable to deploy a computational trust model into real-world collaborative systems.

Limitations of our works are the following:

- We used the factorial experimental design. Due to the nature of the design, we ask same groups of people play different games to compare the user behavior under different contexts.Therefore, it might be difficult in practice to extend the experiment. For instance, it might be difficult to ask a group of people to play six games that might last for three hours.
- We argue and reason that reputation score does not reflect the behavior of users due to the assumption of constant-behavior pattern over partners. The assumption might be verified by organizing an experiment in which we display reputation score to users.

5.1.2 Calculating trust score

We defined the question as, given a log of a collaborative system, for a particular user, how to calculate trust score of each partner who has already interacted with this user.

Because trust score is calculated based on the user contribution to the sharing task, the first step is to calculate the quality of the sharing task. Then we can calculate the contribution by checking how the user behavior affect the quality.

We studied repeated trust game and Wikipedia as two collaborative contexts in designing trust calculation methods.

For repeated trust games:

1. We proposed a novel computational trust method for repeated trust games. The trust model requires only observable information from a user. Therefore, the model does not rely on any external information. Using the model, a user can always calculate trust score of a partner.

2. We validated the trust method against:

   Simulated users: We defined different user types and apply the trust model into these users. We showed that our trust function can punish cheating behavior and distinguish between different user types better than other baseline models.

   Human opinions: We used the dataset provided by [Keser, 2002] in which players in trust game also give rating score (positive, neutral, negative) to their partners. We compare the trust scores calculated by our trust model with human opinion. We showed that our trust model is consistent with human opinion.

   Human behavior: We used the trust scores calculated by our trust model to predict future behavior of users in three trust game datasets: our own dataset, the dataset provided by [Dubois et al., 2012] and the dataset provided by [Bravo et al., 2012]. We showed that our trust model can be used to predict future behavior better than other baseline models.

On the other hand, we presented several methods to measure the quality of articles on Wikipedia which is one of the most important collaborative systems in the world. We also applied the computational trust model presented for trust game into Wikipedia. Our contributions are:

1. We proposed three different methods in measuring quality of Wikipedia articles.

   - We improved state-of-the-art method [Warncke-Wang, Ayukaev, et al., 2015] by introducing new features to the random forest algorithm. We performed a more comprehensive evaluation. We refer this model as random forest based model.
   - We proposed two novel approaches of using deep learning on measuring quality of Wikipedia articles. The first model uses Doc2Vec and Deep Neural Networks (DNN) to predict the quality of articles. The second model uses Recurrent Neural Networks (RNN) with Long-Short Term Memory (LSTM) to predict the quality. We refer these two models as DNN-based and RNN-based model.

   We validated all three models using real-world Wikipedia datasets: English, French and Russian Wikipedia. The random forest based model is available only for English. We showed that the random forest based model and RNN-based model outperform state-of-the-art algorithm in term of *accuracy* and *AUC* scores. The RNN-based model achieves the highest results in predicting, but the cost is longer running time and lack of explanation.

2. We proposed a quality-based trust measurement for Wikipedia coauthors. We applied the trust model that we presented for trust to Wikipedia editors. We used Levenshtein distance as a contribution metric. We considered the quality of Wikipedia articles as a factor to measure the contribution of users. We showed that our trust model can predict the future contributions of users better than other baseline models.

The fact that our single trust model performs better than other baseline methods in both repeated trust game and Wikipedia allows us to expect an application of our trust model in other real-world collaborative systems.

Limitations of our works are the following:

- In repeated trust game, while our model predicts future behavior of senders pretty well, it does not achieve the same performance in predicting future behavior of receivers. We are aware of potential complicated interaction between multiple factors. The trust model can be improved to capture better the behavior trend of receivers.
- In Wikipedia, we proposed to use edit distance as a contribution metric. We might combine our work with previous studies that use edit longevity as contribution metric. For instance, we might use the trust scores calculated by a trust model to predict the survival time of a text.

5.1.3 Predicting trust relations

In large-scale collaborative systems, it is usual that a user needs to interact with a partner that she never interacted with. In this situation, she needs to decide to trust this partner or not. Because there is no previous interaction, the trust model we presented in the previous section is not possible. However, if the partner is not a new member of the system but had interacted and set up the relations with other users, we can predict the trust or distrust relations that the user will have to this partner. Therefore, we can recommend the user to trust or not the partner.

If the information about trust/distrust relations of a subset of users is available, we can predict the future relations which will be established. Because the relations between users can be represented as a signed directed network where vertices are users and edges are their relations, the task of predicting trust/distrust relations became the task of link-sign prediction in the network [Leskovec et al., 2010a; Song and Meyer, 2015].

We presented an approach of using Random Walk, Doc2Vec and RNN-LSTM for predicting the signs of the future links in a network. We used the Wikipedia dataset wherein the users express explicitly their trust/distrust opinions on other users [Leskovec et al., 2010b] as the main testing dataset to validate our solution. We also used Epinions and Slashdot datasets [Leskovec et al., 2010b] for external validation because in fact the solution can be applied in any signed directed network. The experiments showed that our algorithm can predict more accurately than state-of-the-art algorithms in both static and dynamic networks. Furthermore, our algorithm requires only local information, while some existing algorithms ask for full observed network [Leskovec et al., 2010a; Dubois et al., 2012; You et al., 2016].

If the algorithm is deployed into a collaborative system, we can suggest users in collaborating with other users that they do not know. For instance, Alice receives a request from Bob to join a private project on Github, but Alice does not know Bob yet. The algorithm hence can suggest Alice to trust Bob or not.

In order to perform the proposed algorithm, we assumed a trust network where users explicitly express their trust/distrust opinions on other users. The web of trust might not be always available. We might consider other expressions of trust/distrust opinions. For instance, if Alice denies Bob to access her documents, we might consider that Alice does not trust Bob.

To summarize, in this thesis we studied the problem of trust assessment in large-scale collaboration systems from different perspectives. We showed that trust score can be used to encourage the collaboration between users. We designed a trust model to calculate trust scores between users who interacted and verified the model in different contexts. We presented an algorithm to

predict future trust/distrust relations between users who did not interact. Therefore, before a user decides to collaborate or not with a partner, regardless the interaction history between the user and the partner, we can always provide a recommendation to the user to trust the partner or not.

5.2 Perspectives

In this section, we discuss about the limitations of our studies and propose some future research works.

5.2.1 Large-scale trust game experiments

Due to several practical difficulties, we organized trust game experiments with only 30 participants. While the number of participant is not so small compared to other behavioral experiments, we can benefit from large-scale experiments. If we can do so, we can increase the power and confidence of the results.

We can organize large-scale experiments by using crowd-sourcing system like Amazon Turk [N. Zhang, 2010]. We suggest to use oTree [D. L. Chen et al., 2016] which is a web-based tool to deploy behavioral experiments. In our opinion oTree is more suitable than zTree [Fischbacher, 2007] in organizing large-scale experiments, because users do not need to install any client software and can do the experiment over the Internet.

We address several potential research ideas that can be deployed in large-scale user experiments:

Testing the effect of reputation score against trust score. We might test and compare the effect of showing reputation score with the effect of showing trust score to users. We can compare two scores in three aspects: (i) does showing the score have any effect on user behavior; (ii) do users follow the scores; and (iii) can we use the calculated scores to predict future behavior of users.

Testing the effect of showing trust/distrust suggestion based on link-sign prediction on user behavior. We showed that displaying trust scores of partners to users can encourage the collaboration, and we showed that users follow the suggestions of trust scores. However, we did not test yet the effect of showing trust/distrust recommendation to user behavior.

Deploying simulated users to test the reaction of real users. We can deploy different simulated users (but show them as real users to other real users) to see how real users react with some particular kinds of behavior. For instance, we can deploy an honest user who always send a high amount of money, or a cheating user who will deviate at some point in the future.

5.2.2 Validating The Influence of Trust Score in Real-World Systems

We validated the influence of trust score using repeated trust games, i.e. trust score is validated using a non-context lab-control experiment. Indeed, it is very difficult to deploy a computational trust model in an existing system like Github or Google Docs, because we have no right to integrate our idea to these systems. It could take a long time to propose the idea of trust model to the companies that own these systems.

We suggest to deploy the trust model into a collaborative system for education. For instance, MUTE [C.-L. Ignat et al., 2017] is an open-source collaborative editing system developed by COAST team, Inria Nancy Grand-Est, France and can be used for testing the effect of trust score in real-world scenarios.

Wikipedia is used in this thesis because it provides a very well-annotated data and some articles are assigned quality label as ground truth already. For free text editing systems like MUTE, it is more difficult to define the quality of user contribution. [Yim et al., 2017] presented some approaches to determine the quality of a Google Docs document which can be used as a starting point for further study. If one can develop a quality model for MUTE, we can design and deploy a trust model and test that with real users.

5.2.3 Semi-supervised Deep Learning on Networks

We presented a novel approach of using RNN-LSTM for link-sign prediction in dynamic networks. Our algorithm is a supervised algorithm and requires a fully labelled dataset. Studies claimed that a fully labelled network data is not always available. In practice usually we observe a network where only a small part is labelled by users, i.e. a graph where only a subset of its edges are labelled as positive/negative. We suggest to focus on semi-supervised learning algorithms [Kipf and Welling, 2017] in these scenarios.

5.3 Closing Words

Trust is a very important factor not only for the success of collaboration but also in our daily life. In this thesis, we studied trust assessment in large-scale collaboration. We combined knowledge from multiple fields: computer science, psychology and economic to propose trust models for collaborative contexts. We suggested that in measuring trust, what is important is not only the quality of the behavior but also the stability of the behavior. We hope that our contribution can help to understand more about human trust. We expect to see real-world systems to integrate trust models in the future.

Chapter 6

Bibliography

6.1 Publications

Journal Article

Dang, Quang-Vinh and Claudia-Lavinia Ignat (2016d). "Quality assessment of Wikipedia articles: a deep learning approach". In: *ACM SIGWEB Newsletter* 2016.Autumn, p. 5.

Conference Paper

Dang, Quang-Vinh and Claudia-Lavinia Ignat (2016a). "Computational Trust Model for Repeated Trust Games". In: *2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, August 23-26, 2016*. IEEE, pp. 34–41.

— (2016b). "Measuring Quality of Collaboratively Edited Documents: The Case of Wikipedia". In: *2nd IEEE International Conference on Collaboration and Internet Computing, CIC 2016, Pittsburgh, PA, USA, November 1-3, 2016*. IEEE Computer Society, pp. 266–275.

— (2016c). "Performance of real-time collaborative editors at large scale: User perspective". In: *2016 IFIP Networking Conference, Networking 2016 and Workshops, Vienna, Austria, May 17-19, 2016*. IEEE, pp. 548–553.

— (2016e). "Quality Assessment of Wikipedia Articles without Feature Engineering". In: *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, JCDL 2016, Newark, NJ, USA, June 19 - 23, 2016*. Ed. by Nabil R. Adam, Lillian (Boots) Cassel, Yelena Yesha, Richard Furuta, and Michele C. Weigle. ACM, pp. 27–30.

— (2017a). "An end-to-end learning solution for assessing the quality of Wikipedia articles". In: *Proceedings of the 13th International Symposium on Open Collaboration, OpenSym 2017, Galway, Ireland, August 23-25, 2017*. Ed. by Lorraine Morgan. ACM, 4:1–4:10.

— (2017b). "dTrust: A Simple Deep Learning Approach for Social Recommendation". In: *3rd IEEE International Conference on Collaboration and Internet Computing, CIC 2017, San Jose, CA, USA, October 15-17, 2017*. IEEE Computer Society, pp. 209–218.

6.2 Other Publications

Abbass, Hussein A., Garrison W. Greenwood, and Eleni Petraki (2016). "The N-Player Trust Game and its Replicator Dynamics". In: *IEEE Trans. Evolutionary Computation* 20.3, pp. 470–474.

Abdul-Rahman, Alfarez and Stephen Hailes (1997). "A distributed trust model". In: *NSPW*. ACM, pp. 48–60.

Aberer, Karl and Zoran Despotovic (2001). "Managing trust in a peer-2-peer information system". In: *CIKM*. ACM, pp. 310–317.

Adler, B Thomas (2012). "WikiTrust: content-driven reputation for the Wikipedia". In: *PhD thesis, University of California Santa Cruz*.

Adler, B. Thomas and Luca de Alfaro (2007). "A content-driven reputation system for the wikipedia". In: *WWW*. ACM, pp. 261–270.

Adler, B. Thomas, Luca de Alfaro, Ian Pye, and Vishwanath Raman (2008). "Measuring author contributions to the Wikipedia". In: *Int. Sym. Wikis*. ACM.

Adler, B. Thomas, Krishnendu Chatterjee, Luca de Alfaro, Marco Faella, Ian Pye, and Vishwanath Raman (2008). "Assigning trust to Wikipedia content". In: *Int. Sym. Wikis*. ACM.

Adler, Paul S and Clara Xiaoling Chen (2011). "Combining creativity and control: Understanding individual motivation in large-scale collaborative creativity". In: *Accounting, Organizations and Society* 36.2, pp. 63–85.

Agrawal, Priyanka, Vikas K. Garg, and Ramasuri Narayanam (2013). "Link Label Prediction in Signed Social Networks". In: *IJCAI*. IJCAI/AAAI.

Agrawal, Rakshit and Luca de Alfaro (2016). "Predicting the quality of user contributions via LSTMs". In: *OpenSym*. ACM, 19:1–19:10.

Agrawal, Rakshit, Luca de Alfaro, and Vassilis Polychronopoulos (2016). "Learning From Graph Neighborhoods Using LSTMs". In: *CoRR* abs/1611.06882.

Alfaro, Luca de, Ashutosh Kulshreshtha, Ian Pye, and B. Thomas Adler (2011). "Reputation systems for open collaboration". In: *Commun. ACM* 54.8, pp. 81–87.

Allahbakhsh, Mohammad, Aleksandar Ignjatovic, Boualem Benatallah, Seyed-Mehdi-Reza Beheshti, Elisa Bertino, and Norman Foo (2012). "Reputation management in crowdsourcing systems". In: *CollaborateCom*. ICST, pp. 664–671.

Anderhub, Vital, Dirk Engelmann, and Werner Güth (2002). "An experimental study of the repeated trust game with incomplete information". In: *Jour. of Econ. Behavior & Organization* 48.2, pp. 197–216.

Anderka, Maik, Benno Stein, and Nedim Lipka (2012). "Predicting quality flaws in user-generated content: the case of wikipedia". In: *SIGIR*. ACM, pp. 981–990.

Anderson, John R and Lynne M Reder (1999). "The fan effect: New results and new theories". In: *Journal of Experimental Psychology-General* 128.2, pp. 186–197.

Aphinyanaphongs, Yindalon, Lawrence D. Fu, Zhiguo Li, Eric R. Peskin, Efstratios Efstathiadis, Constantin F. Aliferis, and Alexander R. Statnikov (2014). "A comprehensive empirical comparison of modern supervised classification and feature selection methods for text categorization". In: *JASIST* 65.10, pp. 1964–1987.

Arazy, Ofer and Oded Nov (2010). "Determinants of wikipedia quality: the roles of global and local contribution inequality". In: *CSCW*.

Ashraf, Nava, Iris Bohnet, and Nikita Piankov (2006). "Decomposing trust and trustworthiness". In: *Exp. economics* 9.3, pp. 193–208.

Attebury, Ramirose, Julie George, Cindy Judd, and Brad Marcum (2013). "Google docs: a review". In: *Against the Grain* 20.2, p. 9.

Avizienis, Algirdas, Jean-Claude Laprie, Brian Randell, and Carl E. Landwehr (2004). "Basic Concepts and Taxonomy of Dependable and Secure Computing". In: *IEEE Trans. Dependable Sec. Comput.* 1.1, pp. 11–33.

Ba, Sulin and Paul A. Pavlou (2002). "Evidence of the Effect of Trust Building Technology in Electronic markets: Price Premiums and Buyer Behavior". In: *MIS Quarterly* 26.3, pp. 243–268.

Ba, Sulin and Paul A Pavlou (2002). "Evidence of the effect of trust building technology in electronic markets: Price premiums and buyer behavior". In: *MIS quarterly*, pp. 243–268.

Bachi, Giacomo, Michele Coscia, Anna Monreale, and Fosca Giannotti (2012). "Classifying Trust/Distrust Relationships in Online Social Networks". In: *SocialCom/PASSAT*. IEEE, pp. 552–557.

Backstrom, Lars, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna (2012). "Four degrees of separation". In: *WebSci*. ACM, pp. 33–42.

Balliet, Daniel and Paul AM Van Lange (2013). "Trust, conflict, and cooperation: a meta-analysis." In: *Psychological Bulletin* 139.5, p. 1090.

Balouek, Daniel, Alexandra Carpen-Amarie, Ghislain Charrier, Frédéric Desprez, Emmanuel Jeannot, Emmanuel Jeanvoine, Adrien Lèbre, David Margery, Nicolas Niclausse, Lucas Nussbaum, Olivier Richard, Christian Pérez, Flavien Quesnel, Cyril Rohr, and Luc Sarzyniec (2012). "Adding Virtualization Capabilities to the Grid'5000 Testbed". In: *CLOSER (Selected Papers)*. Vol. 367. Communications in Computer and Information Science. Springer, pp. 3–20.

Baran, Nicole M, Paola Sapienza, and Luigi Zingales (2010). *Can we infer social preferences from the lab? Evidence from the trust game*. Tech. rep. National Bureau of Economic Research.

Bays, Hillary (1998). "Framing and face in Internet exchanges: A socio-cognitive approach". In: *Linguistik online* 1.1, pp. 1–11.

Bellemare, Charles and Sabine Kröger (2007). "On representative social capital". In: *European Economic Review* 51.1, pp. 183–202.

Bengio, Yoshua (2009). "Learning Deep Architectures for AI". In: *Found. and Trends in Mac. Learning* 2.1, pp. 1–127.

— (2012). "Practical Recommendations for Gradient-Based Training of Deep Architectures". In: *Neural Networks: Tricks of the Trade (2nd ed.)* Vol. 7700. Lecture Notes in Computer Science. Springer, pp. 437–478.

Ben-Ner, A, L Putterman, and T Ren (2009). *Lavish Returns on Cheap Talk: Non-binding Communication in a Trust Experiment. University of Minnesota*. Tech. rep. Working Papers.

Ben-Ner, Avner and Louis Putterman (2009). "Trust, communication and contracts: An experiment". In: *Jour. of Econ. Behavior & Organization* 70.1, pp. 106–121.

Bente, Gary, Thomas Dratsch, Kai Kaspar, Tabea Häßler, Oliver Bungard, and Ahmad Al-Issa (2014). "Cultures of trust: effects of avatar faces and reputation scores on german and arab players in an online trust-game". In: *PloS one* 9.6, e98297.

Bente, Gary, Thomas Dratsch, Simon Rehbach, Matthias Reyl, and Blerta Lushaj (2014). "Do You Trust My Avatar? Effects of Photo-Realistic Seller Avatars and Reputation Scores on Trust in Online Transactions". In: *HCI (18)*. Vol. 8527. Lecture Notes in Computer Science. Springer, pp. 461–470.

Bente, Gary, Sabine Rüggenberg, Nicole C Krämer, and Felix Eschenburg (2008). "Avatar-Mediated Networking: Increasing Social Presence and Interpersonal Trust in Net-Based Collaborations". In: *Human communication research* 34.2, pp. 287–318.

Benz, Matthias and Stephan Meier (2008). "Do people behave in experiments as in the field?—evidence from donations". In: *Experimental economics* 11.3, pp. 268–281.

Berg, Joyce, John Dickhaut, and Kevin McCabe (1995). "Trust, reciprocity, and social history". In: *Games and economic behavior* 10.1, pp. 122–142.

Bergstra, James and Yoshua Bengio (2012). "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13, pp. 281–305.

Betancourt, Grace Gimon, Armando Segnine, Carlos Trabuco, Amira Rezgui, and Nicolas Jullien (2016). "Mining team characteristics to predict Wikipedia article quality". In: *OpenSym.* ACM, 15:1–15:9.

Bhargav-Spantzel, Abhilasha, Jungha Woo, and Elisa Bertino (2007). "Receipt management-transaction history based trust establishment". In: *Digital Identity Management.* ACM, pp. 82–91.

Biancani, Susan (2014). "Measuring the Quality of Edits to Wikipedia". In: *OpenSym.* ACM, 33:1–33:3.

Blumenstock, Joshua Evan (2008). "Size matters: word count as a measure of quality on wikipedia". In: *WWW.* ACM, pp. 1095–1096.

Bohnet, Iris and Richard Zeckhauser (2004). "Trust, risk and betrayal". In: *Jour. of Econ. Behavior & Organization* 55.4, pp. 467–484.

Böhning, Dankmar (1992). "Multinomial logistic regression algorithm". In: *Annals of the Institute of Statistical Mathematics* 44.1, pp. 197–200.

Bolton, Gary E, Elena Katok, and Axel Ockenfels (2002). "How effective are online reputation mechanisms? An experimental investigation". In: *Management Science.* Citeseer.

Bolton, Gary E., Elena Katok, and Axel Ockenfels (2005). "Cooperation among strangers with limited information about reputation". In: *Jour. of Pub. Econ.* 89.8, pp. 1457–1468.

Bornhorst, Fabian, Andrea Ichino, Oliver Kirchkamp, Karl H. Schlag, and Eyal Winter (2010). "Similarities and differences when building trust: the role of cultures". In: *Exp. Econ.* 13.3, pp. 260–283.

Bourgeois-Gironde, Sacha and Anne Corcos (2011). "Discriminating strategic reciprocity and acquired trust in the repeated trust-game". In: *Econ. Bulletin* 31.1, pp. 177–188.

Bracht, Juergen and Nick Feltovich (2008). "Efficiency in the trust game: an experimental study of precommitment". In: *International Jour. of Game Theory* 37.1, pp. 39–72.

— (2009). "Whatever you say, your reputation precedes you: Observation and cheap talk in the trust game". In: *Jour. of Pub. Econ.* 93.9, pp. 1036–1044.

Branco, Paula, Luís Torgo, and Rita P. Ribeiro (2016). "A Survey of Predictive Modeling on Imbalanced Domains". In: *ACM Comput. Surv.* 49.2, 31:1–31:50.

Brandenburger, Adam M and Barry J Nalebuff (2011). *Co-opetition.* Crown Business.

Bravo, Giangiacomo, Flaminio Squazzoni, and Riccardo Boero (2012). "Trust and partner selection in social networks: An experimentally grounded model". In: *Social Networks* 34.4, pp. 481–492.

Braynov, Sviatoslav and Tuomas Sandholm (2002). "Contracting with uncertain level of trust". In: *Comput. Intelli.* 18.4, pp. 501–514.

Breitmoser, Yves (2015). "Cooperation, but no reciprocity: Individual strategies in the repeated Prisoner's Dilemma". In: *The American Economic Review* 105.9, pp. 2882–2910.

Brown, Adam R (2011). "Wikipedia as a data source for political scientists: Accuracy and completeness of coverage". In: *PS: Political Science & Politics* 44.02, pp. 339–343.

Brülhart, Marius and Jean-Claude Usunier (2012). "Does the trust game measure trust?" In: *Econ. Letters* 115.1, pp. 20–23.

Bruttel, Lisa and Ulrich Kamecke (2012). "Infinity in the lab. How do people play repeated games?" In: *Theory and Decision* 72.2, pp. 205–219.

Buchan, Nancy R, Rachel TA Croson, and Sara Solnick (2008). "Trust and gender: An examination of behavior and beliefs in the Investment Game". In: *Journal of Economic Behavior & Organization* 68.3, pp. 466–476.

Buntain, Cody and Jennifer Golbeck (2015). "Trust transfer between contexts". In: *Journal of Trust Management* 2.1, 6.

Burke, Moira and Robert E. Kraut (2008). "Mopping up: modeling wikipedia promotion decisions". In: *CSCW*. ACM, pp. 27–36.

Burks, Stephen V, Jeffrey P Carpenter, and Eric Verhoogen (2003). "Playing both roles in the trust game". In: *Journal of Economic Behavior & Organization* 51.2, pp. 195–216.

Butler, Jeffrey V, Paola Giuliano, and Luigi Guiso (2016). "Trust and cheating". In: *Economic Journal*, pp. 1703–1740.

Calzada, Gabriel De la and Alex Dekhtyar (2010). "On measuring the quality of Wikipedia articles". In: *WICOW*. ACM, pp. 11–18.

Camera, Gabriele and Marco Casari (2009). "Cooperation among strangers under the shadow of the future". In: *The American Economic Review* 99.3, pp. 979–1005.

Camerer, Colin (2003). *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press.

Castelvecchi, D (2015). *Physics paper sets record with more than 5000 authors. Nature News*.

Center, Dropbox Help (2017). *How many people can edit and use Dropbox Paper at one time?* URL: https://www.dropboxforum.com/t5/Dropbox-Paper/How-many-people-can-edit-and-use-Dropbox-Paper-at-one-time/td-p/208053 (visited on 07/03/2017).

Cesarini, David, Christopher T Dawes, James H Fowler, Magnus Johannesson, Paul Lichtenstein, and Björn Wallace (2008). "Heritability of cooperative behavior in the trust game". In: *Proceedings of the National Academy of sciences* 105.10, pp. 3721–3726.

Chacon, Scott and Ben Straub (2014). *Pro git*. Apress.

Chakravarty, Sujoy, Daniel Friedman, Gautam Gupta, Neeraj Hatekar, Santanu Mitra, and Shyam Sunder (2011). "Experimental economics: a survey". In: *Economic & Political Weekly* 46.35, p. 39.

Chall, Jeanne Sternlicht and Edgar Dale (1995). *Readability revisited: The new Dale-Chall readability formula*. Brookline Books.

Chandrashekar, Girish and Ferat Sahin (2014). "A survey on feature selection methods". In: *Computers & Electrical Engineering* 40.1, pp. 16–28.

Charness, Gary and Uri Gneezy (2008). "What's in a name? Anonymity and social distance in dictator and ultimatum games". In: *Journal of Economic Behavior & Organization* 68.1, pp. 29–35.

Charness, Gary and Peter Kuhn (2011). "Lab labor: What can labor economists learn from the lab?" In: *Handbook of labor economics* 4, pp. 229–330.

Chen, Bo and Reza Curtmola (2014). "Auditable Version Control Systems." In: *NDSS*.

Chen, Daniel L, Martin Schonger, and Chris Wickens (2016). "oTree—An open-source platform for laboratory, online, and field experiments". In: *Journal of Behavioral and Experimental Finance* 9, pp. 88–97.

Chen, Hai-Hon (2012). "How to Use Readability Formulas to Access and Select English Reading Materials." In: *Journal of Educational Media & Library Sciences* 50.2.

Chen, Xi and Shuo Shi (2009). "A literature review of privacy research on social network sites". In: *MINES*. Vol. 1. IEEE, pp. 93–97.

Chetlur, Sharan, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer (2014). "cuDNN: Efficient Primitives for Deep Learning". In: *CoRR* abs/1410.0759.

Chiang, Kai-Yang, Nagarajan Natarajan, Ambuj Tewari, and Inderjit S. Dhillon (2011). "Exploiting longer cycles for link prediction in signed networks". In: *CIKM*. ACM, pp. 1157–1162.

Cho, Jin-Hee, Kevin S. Chan, and Sibel Adali (2015). "A Survey on Trust Modeling". In: *ACM Comput. Surv.* 48.2, p. 28.

Choi, Dongho, Chirag Shah, and Vivek Singh (2016). "Which team benefits from collaboration?: Investigating collaborative information seeking using personal and social contextual signals". In: *Proceedings of the Association for Information Science and Technology* 53.1, pp. 1–6.

Chung, Joon Son, Andrew W. Senior, Oriol Vinyals, and Andrew Zisserman (2016). "Lip Reading Sentences in the Wild". In: *CoRR* abs/1611.05358.

Chung, Junyoung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio (2014). "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *CoRR* abs/1412.3555.

Clemons, Eric K., Joshua S. Wilson, Christian Matt, Thomas Hess, Fei Ren, and Fujie Jin (2016). "Online Trust: An International Study of Subjects' Willingness to Shop at Online Merchants, Including the Effects of Promises and of Third Party Guarantees". In: *HICSS*. IEEE Computer Society, pp. 5220–5229.

Cochard, Francois, Phu Nguyen Van, and Marc Willinger (2004). "Trusting behavior in a repeated investment game". In: *Jour. of Econ. Behavior & Organization* 55.1, pp. 31–44.

Cohen, Susan G and Don Mankin (1999). "Collaboration in the virtual organization". In: *Journal of Organizational Behavior* 6, p. 105.

Coleman, Meri and Ta Lin Liau (1975). "A computer readability formula designed for machine scoring." In: *Journal of Applied Psychology* 60.2, p. 283.

Colombo, Ferdinando and Guido Merzoni (2006). "In praise of rigidity: The bright side of long-term contracts in repeated trust games". In: *Jour. of Econ. Behavior & Organization* 59.3, pp. 349–373.

Connor, Jerome T., Douglas R. Martin, and Les E. Atlas (1994). "Recurrent neural networks and robust time series prediction". In: *IEEE Trans. Neural Networks* 5.2, pp. 240–254.

Cooper, David and John H Kagel (2016). "Other regarding preferences: a selective survey of experimental results". In: *Handbook of experimental economics*. Vol. 2, pp. 217–289.

Corbitt, Brian J., Theerasak Thanasankit, and Han Yi (2003). "Trust and e-commerce: a study of consumer perceptions". In: *Electronic Commerce Research and Applications* 2.3, pp. 203–215.

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (2009). *Introduction to Algorithms (3. ed.)* MIT Press.

Covington, Paul, Jay Adams, and Emre Sargin (2016). "Deep Neural Networks for YouTube Recommendations". In: *RecSys*. ACM, pp. 191–198.

Crawley, Michael J (2015). *Statistics: an introduction using R*. 2nd ed. John Wiley & Sons Ltd.

Cruz, Guilherme A. Maldonado da, Elisa Hatsue Moriya Huzita, and Valéria Delisandra Feltrim (2016). "Estimating Trust in Virtual Teams - A Framework based on Sentiment Analysis". In: *ICEIS (1)*. SciTePress, pp. 464–471.

Cygan, Marek, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk (2015). "Sitting Closer to Friends than Enemies, Revisited". In: *Theory Comput. Syst.* 56.2, pp. 394–405.

Dale, Edgar and Jeanne S Chall (1948). "A formula for predicting readability: Instructions". In: *Educational research bulletin*, pp. 37–54.

Dalip, Daniel Hasan, Marcos André Goncalves, Marco Cristo, and Pável Calado (2009). "Automatic quality assessment of content created collaboratively by web communities: a case study of Wikipedia". In: *JCDL*. ACM, pp. 295–304.

Dalip, Daniel Hasan, Harlley Lima, Marcos André Goncalves, Marco Cristo, and Pável Calado (2014). "Quality assessment of collaborative content with minimal information". In: *JCDL*.

Dang, Quang-Vinh (2017). "Analyzing the trust game experiments". In: *DOI: 10.5281/zenodo.802139*.

Das, Anupam and Mohammad Mahfuzul Islam (2012). "SecuredTrust: A Dynamic Trust Computation Model for Secured Communication in Multiagent Systems". In: *IEEE Trans. Dependable Sec. Comput.* 9.2, pp. 261–274.

Dasgupta, Partha (2000). "Trust as a Commodity". In: *Trust: Making and Breaking Cooperative Relations*. Ed. by Diego Gambetta. University of Oxford, pp. 49–72.

Dean, Jeffrey and Sanjay Ghemawat (2008). "MapReduce: simplified data processing on large clusters". In: *Commun. ACM* 51.1, pp. 107–113.

Delgado-Márquez, Blanca L, Nuria E Hurtado-Torres, and J Alberto Aragón-Correa (2012). "The dynamic nature of trust transfer: Measurement and the influence of reciprocity". In: *Decision Support Systems* 54.1, pp. 226–234.

Deng, Li and Dong Yu (2014). "Deep Learning: Methods and Applications". In: *Foundations and Trends in Signal Processing* 7.3-4, pp. 197–387.

Deng, S., L. Huang, G. Xu, X. Wu, and Z. Wu (2016). "On Deep Learning for Trust-Aware Recommendations in Social Networks". In: *IEEE TNNLS* PP.99, pp. 1–14.

Denning, Dorothy E. (1993). "A new paradigm for trusted systems". In: *NSPW*. ACM, pp. 36–41.

Dewan, Prasun and Rajiv Choudhary (1991). "Flexible user interface coupling in a collaborative system". In: *CHI*. ACM, pp. 41–48.

Dhamija, Rachna and Adrian Perrig (2000). "Deja Vu-A User Study: Using Images for Authentication". In: *USENIX Security Symposium*. USENIX Association.

Dinesen, Peter Thisted and Rene Bekkers (2015). "The Foundations of Individuals' Generalized Social Trust: A Review". In: *Trust in Social Dilemmas*. Human Cooperation. Oxford University Press, p. 30.

Dinger, Jochen and Hannes Hartenstein (2006). "Defending the Sybil Attack in P2P Networks: Taxonomy, Challenges, and a Proposal for Self-Registration". In: *ARES*. IEEE Computer Society, pp. 756–763.

Dix, Alan (2009). *Human-computer interaction*. Springer.

Doan, AnHai, Raghu Ramakrishnan, and Alon Y Halevy (2010). "Mass collaboration systems on the world-wide web". In: *Communications of the ACM* 54.4, pp. 86–96.

Doney, Patricia M and Joseph P Cannon (1997). "An examination of the nature of trust in buyer-seller relationships". In: *the Journal of Marketing*, pp. 35–51.

Dong, Yuxiao, Jie Tang, Sen Wu, Jilei Tian, Nitesh V. Chawla, Jinghai Rao, and Huanhuan Cao (2012). "Link Prediction and Recommendation across Heterogeneous Social Networks". In: *ICDM*. IEEE Computer Society, pp. 181–190.

Douceur, John R. (2002). "The Sybil Attack". In: *IPTPS*. Vol. 2429. Lecture Notes in Computer Science. Springer, pp. 251–260.

Duan, Wenjing, Bin Gu, and Andrew B. Whinston (2008). "Do online reviews matter? - An empirical investigation of panel data". In: *Decision Support Systems* 45.4, pp. 1007–1016.

Dubois, Dimitri, Marc Willinger, and Thierry Blayac (2012). "Does players' identification affect trust and reciprocity in the lab?" In: *Jour. of Econ. Psychology* 33.1, pp. 303–317.

DuBois, Thomas, Jennifer Golbeck, and Aravind Srinivasan (2011). "Predicting Trust and Distrust in Social Networks". In: *SocialCom/PASSAT*. IEEE, pp. 418–424.

Easley, David A. and Jon M. Kleinberg (2010). *Networks, Crowds, and Markets - Reasoning About a Highly Connected World*. Cambridge University Press.

Economist, The (2016). *Why research papers have so many authors*. URL: http://www.economist.com/news/science-and-technology/21710792-scientific-publications-are-getting-more-and-more-names-attached-them-why (visited on 05/18/2017).

Edunov, Sergey, Carlos Diuk, Ismail Onur Filiz, Smriti Bhagat, and Moira Burke (2016). "Three and a half degrees of separation". In: *Research at Facebook*.

Edwards, Jennifer T (2011). "A Case Study: Using Google Documents as a Collaborative Writing Tool in Undergraduate Courses". In: *The Texas Speech Communication Journal Online*.

Elliott, Mark Alan (2007). "Stigmergic collaboration: A theoretical framework for mass collaboration". PhD thesis. University of Melbourne.

Engle-Warnick, Jim and Robert L Slonim (2001). *The Fragility and Robustness of Trust*. Tech. rep. Econ. Group, Nuffield College, University of Oxford.

Engle-Warnick, Jim and Robert L. Slonim (2004). "The evolution of strategies in a repeated trust game". In: *Jour. of Econ. Behavior & Organization* 55.4, pp. 553–573.

Engle-Warnick, Jim and Robert L Slonim (2006). "Inferring repeated-game strategies from actions: evidence from trust game experiments". In: *Econ. theory* 28.3, pp. 603–632.

Engle-Warnick, Jim and Robert L. Slonim (2006). "Learning to trust in indefinitely repeated games". In: *Games and Econ. Behavior* 54.1, pp. 95–114.

Erickson, Tamara J and L Gratton (2007). "Eight ways to build collaborative teams". In: *Harvard business review* 11, pp. 1–11.

Ert, Eyal, Aliza Fleischer, and Nathan Magen (2016). "Trust and reputation in the sharing economy: The role of personal photos in Airbnb". In: *Tourism Management* 55, pp. 62–73.

Evans, Anthony M and William Revelle (2008). "Survey and behavioral measurements of interpersonal trust". In: *Jour. of Research in Personality* 42.6, pp. 1585–1593.

Even, Shimon (2011). *Graph algorithms*. 2nd ed. Cambridge University Press.

Falk, Armin and James J Heckman (2009). "Lab experiments are a major source of knowledge in the social sciences". In: *Science* 326.5952, pp. 535–538.

Falk, Armin, Stephan Meier, and Christian Zehnder (2010). "Did we overestimate the role of social preferences? The case of self-selected student samples". In: *CESifo Working Paper Series*.

Fehr, Ernst, Urs Fischbacher, Bernhard Von Rosenbladt, Jürgen Schupp, and Gert G Wagner (2003). "A nation-wide laboratory: examining trust and trustworthiness by integrating behavioral experiments into representative survey". In: *CESifo Working Paper Series*.

Feltovich, Nick and Joe Swierzbinski (2011). "The role of strategic uncertainty in games: An experimental study of cheap talk and contracts in the Nash demand game". In: *European Econ. Review* 55.4, pp. 554–574.

Fetchenhauer, Detlef and David Dunning (2009). "Do people trust too much or too little?" In: *Journal of Economic Psychology* 30.3, pp. 263–276.

Firestine, Brandi (2017). *Celebrating nine years of GitHub with an anniversary sale*. URL: https://github.com/blog/2345-celebrating-nine-years-of-github-with-an-anniversary-sale (visited on 07/03/2017).

Fischbacher, Urs (2007). "z-Tree: Zurich toolbox for ready-made economic experiments". In: *Exp. economics* 10.2, pp. 171–178.

Forte, Andrea and Amy Bruckman (2005). "Why do people write for Wikipedia? Incentives to contribute to open–content publishing". In: *GROUP* 5, pp. 6–9.

Foundation, Wikimedia (2015). *Objective Revision Evaluation Service*. URL: https://meta.wikimedia.org/wiki/Objective_Revision_Evaluation_Service (visited on 12/01/2015).

Franklin Jr, Curtis F (1997). "Emerging Technology: Enter the Extranet". In: *CIO Magazine* 15.

Fredrickson, Barbara L and Daniel Kahneman (1993). "Duration neglect in retrospective evaluations of affective episodes". In: *Jour. of personality and social psychology* 65.1, p. 45.

Gal, Yarin and Zoubin Ghahramani (2016). "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks". In: *NIPS*, pp. 1019–1027.

Galar, Mikel, Alberto Fernández, Edurne Barrenechea Tartas, Humberto Bustince Sola, and Francisco Herrera (2012). "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches". In: *IEEE Trans. Systems, Man, and Cybernetics, Part C* 42.4, pp. 463–484.

Gambetta, Diego (1988). "Can we trust trust". In: *Trust: Making and breaking cooperative relations*. Basil Blackwell, pp. 213–237.

Gaver, William W. and Randall B. Smith (1990). "Auditory icons in large-scale collaborative environments". In: *INTERACT*. North-Holland, pp. 735–740.

Gerber, Adam and Clifton Craig (2015). "Introducing Git". In: *Learn Android Studio*. Springer, pp. 145–187.

Gers, Felix A., Jürgen Schmidhuber, and Fred A. Cummins (2000). "Learning to Forget: Continual Prediction with LSTM". In: *Neural Computation* 12.10, pp. 2451–2471.

Glaeser, Edward L., David I. Laibson, Jose A. Scheinkman, and Christine L. Soutter (2000). "Measuring trust". In: *Quarterly Jour. of Econ.* Pp. 811–846.

Gobby (2017). *Gobby: a collaborative text editor*. URL: https://gobby.github.io (visited on 05/15/2017).

Golbeck, Jennifer (2009). "Introduction to Computing with Social Trust". In: *Computing with Social Trust*. Human-Computer Interaction Series. Springer, pp. 1–5.

Golbeck, Jennifer and James A. Hendler (2006). "Inferring binary trust relationships in Web-based social networks". In: *ACM Trans. Internet Techn.* 6.4, pp. 497–529.

Gollapudi, Sunila (2016). *Practical Machine Learning*. Packt Publishing Ltd.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. MIT Press.

Goodwin, Danny (2012). *Wikipedia Appears on Page 1 of Google for 99% of Searches*. URL: https://searchenginewatch.com/sew/study/2152194/wikipedia-google-search-results-study (visited on 05/29/2017).

Google (2017). *Share files from Google Drive*. URL: https://support.google.com/docs/answer/2494822?hl=en&visit_id=1-636306298860663307-4197237845&rd=2 (visited on 05/17/2017).

Grabner-Kraeuter, Sonja (2002). "The role of consumers' trust in online-shopping". In: *Journal of Business Ethics* 39.1-2, pp. 43–50.

Granatyr, Jones, Vanderson Botelho, Otto Robert Lessing, Edson Emílio Scalabrin, Jean-Paul A. Barthès, and Fabrício Enembreck (2015). "Trust and Reputation Models for Multiagent Systems". In: *ACM Comput. Surv.* 48.2, p. 27.

Grave, Edouard, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou (2016). "Efficient softmax approximation for GPUs". In: *CoRR* abs/1609.04309.

Graves, Alex (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*. Vol. 385. Studies in Computational Intelligence. Springer.

— (2013). "Generating Sequences With Recurrent Neural Networks". In: *CoRR* abs/1308.0850.

Greenhalgh, Christopher (1997). "Large scale collaborative virtual environments". PhD thesis. University of Nottingham.

Grossklags, Jens (2007). "Experimental economics and experimental computer science: a survey". In: *Experimental Computer Science*. ACM, p. 11.

Grover, Aditya and Jure Leskovec (2016). "node2vec: Scalable Feature Learning for Networks". In: *KDD*. ACM, pp. 855–864.

Gu, Ning, Qiwei Zhang, Jiangming Yang, and Wei Ye (2007). "Dcv: a causality detection approach for large-scale dynamic collaboration environments". In: *GROUP*. ACM, pp. 157–166.

Guare, John (1990). *Six degrees of separation: A play*. Vintage.

Guha, Ramanathan V., Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins (2004). "Propagation of trust and distrust". In: *WWW*. ACM, pp. 403–412.

Gunning, Robert (1969). "The fog index after twenty years". In: *Journal of Business Communication* 6.2, pp. 3–13.

Gunnthorsdottir, Anna, Kevin McCabe, and Vernon Smith (2002). "Using the Machiavellianism instrument to predict trustworthiness in a bargaining game". In: *Jour. of Econ. Psychology* 23.1, pp. 49–66.

Guyon, Isabelle and André Elisseeff (2003). "An Introduction to Variable and Feature Selection". In: *Journal of Machine Learning Research* 3, pp. 1157–1182.

Halfaker, Aaron, Aniket Kittur, and John Riedl (2011). "Don't bite the newbies: how reverts affect the quantity and quality of Wikipedia work". In: *Int. Sym. Wikis*. ACM, pp. 163–172.

Halfaker, Aaron and Dario Taraborelli (2015). *Artificial intelligence service gives Wikipedians 'X-ray specs' to see through bad edits*. URL: https://blog.wikimedia.org/2015/11/30/artificial-intelligence-x-ray-specs (visited on 05/11/2017).

Hand, David J. and Robert J. Till (2001). "A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems". In: *Machine Learning* 45.2, pp. 171–186.

Harris, Zellig S (1954). "Distributional structure." In: *Word*.

Haselhuhn, Michael P, Jessica A Kennedy, Laura J Kray, Alex B Van Zant, and Maurice E Schweitzer (2015). "Gender differences in trust dynamics: Women trust more than men following a trust violation". In: *Journal of Experimental Social Psychology* 56, pp. 104–109.

Hemp, Paul (2006). "Avatar-based marketing". In: *Harvard business review* 84.6, pp. 48–57.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780.

Hoelz, Bruno W. P. and Célia Ghedini Ralha (2015). "Towards a cognitive meta-model for adaptive trust and reputation in open multi-agent systems". In: *Auto. Agents and Mul.-Agent Systems* 29.6, pp. 1125–1156.

Hoffman, Kevin J., David Zage, and Cristina Nita-Rotaru (2009). "A survey of attack and defense techniques for reputation systems". In: *ACM Comput. Surv.* 42.1.

Hsieh, Cho-Jui, Kai-Yang Chiang, and Inderjit S. Dhillon (2012). "Low rank modeling of signed networks". In: *KDD*.

Hu, Meiqun, Ee-Peng Lim, Aixin Sun, Hady Wirawan Lauw, and Ba-Quy Vuong (2007). "Measuring article quality in Wikipedia: models and evaluation". In: *CIKM*. ACM, pp. 243–252.

Huang, Jin and Charles X. Ling (2005). "Using AUC and Accuracy in Evaluating Learning Algorithms". In: *IEEE Trans. Knowl. Data Eng.* 17.3, pp. 299–310.

Hupfer, Susanne, Li-Te Cheng, Steven I. Ross, and John F. Patterson (2004). "Introducing collaboration into an application development environment". In: *CSCW*. ACM, pp. 21–24.

Huynh, Trung Dong (2009). "A personalized framework for trust assessment". In: *International Symposium on Applied Computing*. ACM, pp. 1302–1307.

Ignat, Claudia-Lavinia, Luc André, and Gérald Oster (2017). "Enhancing rich content wikis with real-time collaboration". In: *Concurrency and Computation: Practice and Experience*.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.

Jang, Hyunju, Kihong Kim, Sun Huh, et al. (2016). "Increasing number of authors per paper in Korean science and technology papers". In: *Science Editing* 3.2, pp. 80–89.

Japkowicz, Nathalie and Mohak Shah, eds. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press.

Javanmardi, Sara, Yasser Ganjisaffar, Cristina Videira Lopes, and Pierre Baldi (2009). "User contribution and trust in Wikipedia". In: *CollaborateCom*. ICST, pp. 1–6.

Javanmardi, Sara and Cristina Lopes (2010). "Statistical Measure of Quality in Wikipedia". In: *Proceedings of the First Workshop on Social Media Analytics*. SOMA '10. Washington D.C., District of Columbia: ACM, pp. 132–138.

Javanmardi, Sara, Cristina Videira Lopes, and Pierre Baldi (2010). "Modeling user reputation in wikis". In: *Statistical Analysis and Data Mining* 3.2, pp. 126–139.

Jiang, Wenjun, Guojun Wang, Md. Zakirul Alam Bhuiyan, and Jie Wu (2016). "Understanding Graph-Based Trust Evaluation in Online Social Networks: Methodologies and Challenges". In: *ACM Comput. Surv.* 49.1, p. 10.

Johansson-Stenman, Olof, Minhaj Mahmud, and Peter Martinsson (2013). "Trust, trust games and stated trust: Evidence from rural Bangladesh". In: *Journal of Economic Behavior & Organization* 95, pp. 286–298.

Johnson, Noel D and Alexandra A Mislin (2011). "Trust games: A meta-analysis". In: *Journal of Economic Psychology* 32.5, pp. 865–889.

Jones, S (2009). "The social life of health information". In: *Pew research center, Washington, DC, Pew Internet & American Life Project*.

Jøsang, Audun, John Fabre, Brian Hay, James Dalziel, and Simon Pope (2005). "Trust Requirements in Identity Management". In: *ACSW Frontiers*. Vol. 44. CRPIT. Australian Computer Society, pp. 99–108.

Jøsang, Audun and Jennifer Golbeck (2009). "Challenges for robust trust and reputation systems". In: *STM*.

Jøsang, Audun, Roslan Ismail, and Colin Boyd (2007). "A Survey of Trust and Reputation Systems for Online Service Provision". In: *Decis. Support Syst.* 43.2, pp. 618–644.

Jøsang, Audun, Stephen Marsh, and Simon Pope (2006). "Exploring Different Types of Trust Propagation". In: *iTrust*. Vol. 3986. Lecture Notes in Computer Science. Springer, pp. 179–192.

Józefowicz, Rafal, Wojciech Zaremba, and Ilya Sutskever (2015). "An Empirical Exploration of Recurrent Network Architectures". In: *ICML*. Vol. 37. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 2342–2350.

Kanji, Gopal K (2006). *100 statistical tests*. 3rd ed. Sage.

Karlan, Dean S (2005). "Using experimental economics to measure social capital and predict financial decisions". In: *The American economic review* 95.5, pp. 1688–1699.

Kasper-Fuehrera, Eva C and Neal M Ashkanasy (2001). "Communicating trustworthiness and building trust in interorganizational virtual organizations". In: *Journal of management* 27.3, pp. 235–254.

Kendall, Graham, Xin Yao, and Siang Yew Chong (2007). *The Iterated Prisoners' Dilemma - 20 Years On*. Vol. 4. Advances in Natural Computation. World Scientific.

Keppel, Geoffrey (1991). *Design and analysis: A researcher's handbook* . Prentice-Hall, Inc.

Keser, Claudia (2002). *Trust and reputation building in e-commerce.* Tech. rep. IBM Watson Research Center working paper.

Keser, Claudia (2003). "Trust in a Networked World: Experimental Games for the Design of Reputation Management Systems". In: *IBM Watson Working Report*.

Kincaid, J Peter, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom (1975). *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel*. Tech. rep. DTIC Document.

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980.

Kipf, Thomas N. and Max Welling (2017). "Semi-Supervised Classification with Graph Convolutional Networks". In: *ICLR*, p. 14.

Kittur, Aniket and Robert E. Kraut (2008). "Harnessing the wisdom of crowds in Wikipedia: quality through coordination". In: *CSCW*. ACM, pp. 37–46.

Kolowich, Steve (2013). *Georgia Tech and Coursera Try to Recover From MOOC Stumble*. URL: http://www.chronicle.com/blogs/wiredcampus/georgia-tech-and-coursera-try-to-recover-from-mooc-stumble/42167 (visited on 04/21/2017).

Kraut, Robert, Jolene Galegher, Robert Fish, and Barbara Chalfonte (1992). "Task requirements and media choice in collaborative writing". In: *Human–Computer Interaction* 7.4, pp. 375–407.

Kuznetsov, Stacey (2006). "Motivations of contributors to Wikipedia". In: *SIGCAS Computers and Society* 36.2, p. 1.

La Robertie, Baptiste de, Yoann Pitarch, and Olivier Teste (2015). "Measuring Article Quality in Wikipedia using the Collaboration Network". In: *ASONAM*. ACM, pp. 464–471.

Laaksonen, Toni, Toni Jarimo, and Harri I Kulmala (2009). "Cooperative strategies in customer–supplier relationships: The role of interfirm trust". In: *International Journal of Production Economics* 120.1, pp. 79–87.

Laniado, David and Riccardo Tasso (2011). "Co-authorship 2.0: patterns of collaboration in Wikipedia". In: *HT*. ACM, pp. 201–210.

Le, Quoc V. and Tomas Mikolov (2014). "Distributed Representations of Sentences and Documents". In: *ICML*. Vol. 32. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 1188–1196.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *Nature* 521.7553, pp. 436–444.

Lee, Jumin, Do-Hyung Park, and Ingoo Han (2008). "The effect of negative online consumer reviews on product attitude: An information processing view". In: *Electronic Commerce Research and Applications* 7.3, pp. 341–352.

Lee, Matthew K. O. and Efraim Turban (2001). "A Trust Model for Consumer Internet Shopping". In: *Int. J. Electronic Commerce* 6.1, pp. 75–91.

Leskovec, Jure and Christos Faloutsos (2006). "Sampling from large graphs". In: *KDD*. ACM, pp. 631–636.

Leskovec, Jure, Daniel P. Huttenlocher, and Jon M. Kleinberg (2010a). "Predicting positive and negative links in online social networks". In: *WWW*. ACM, pp. 641–650.

— (2010b). "Signed networks in social media". In: *CHI*. ACM, pp. 1361–1370.

Lesmeister, Cory (2015). *Mastering Machine Learning with R*. Packt Publishing Ltd.

Levenshtein, Vladimir I (1966). "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics doklady*. Vol. 10. 8, pp. 707–710.

Levitt, Steven D and John A List (2009). "Field experiments in economics: The past, the present, and the future". In: *European Economic Review* 53.1, pp. 1–18.

Li, Rong-Hua, Jeffrey Xu Yu, Lu Qin, Rui Mao, and Tan Jin (2015). "On random walk based graph sampling". In: *ICDE*. IEEE Computer Society, pp. 927–938.

Li, Xiaoyi, Nan Du, Hui Li, Kang Li, Jing Gao, and Aidong Zhang (2014). "A Deep Learning Approach to Link Prediction in Dynamic Networks". In: *SDM*. SIAM, pp. 289–297.

Li, Xinyi, Jintao Tang, Ting Wang, Zhunchen Luo, and Maarten de Rijke (2015). "Automatically Assessing Wikipedia Article Quality by Exploiting Article-Editor Networks". In: *ECIR*. Vol. 9022. Lecture Notes in Computer Science, pp. 574–580.

Liben-Nowell, David and Jon M. Kleinberg (2007). "The link-prediction problem for social networks". In: *JASIST*.

Lipka, Nedim and Benno Stein (2010). "Identifying featured articles in wikipedia: writing style matters". In: *WWW*. ACM, pp. 1147–1148.

List, John, Steven Levitt, et al. (2007). *What do laboratory experiments measuring social preferences reveal about the real world*. Tech. rep. The Field Experiments Website.

Liu, Feng, Bingquan Liu, Chengjie Sun, Ming Liu, and Xiaolong Wang (2013). "Deep Learning Approaches for Link Prediction in Social Network Services". In: *ICONIP*, pp. 425–432.

Liu, Jun and Sudha Ram (2011). "Who does what: Collaboration patterns in the Wikipedia and their impact on article quality". In: *ACM Trans. Management Inf. Syst.* 2.2, p. 11.

Liu, Xin, Anwitaman Datta, and Krzysztof Rzadca (2013). "Trust beyond reputation: A computational trust model based on stereotypes". In: *Electronic Commerce Research and Applications* 12.1, pp. 24–39.

Lunawat, Radhika (2013). "An experimental investigation of reputation effects of disclosure in an investment/trust game". In: *Journal of Economic Behavior & Organization* 94, pp. 130–144.

Lv, Xiao, Fazhi He, Weiwei Cai, and Yuan Cheng (2016). "A string-wise CRDT algorithm for smart and large-scale collaborative editing systems". In: *Advanced Engineering Informatics*.

Malaga, Ross A. (2001). "Web-Based Reputation Management Systems: Problems and Suggested Solutions". In: *Electronic Commerce Research* 1.4, pp. 403–417.

Maniu, Silviu, Bogdan Cautis, and Talel Abdessalem (2011). "Building a signed network from interactions in Wikipedia". In: *DBSocial*. ACM, pp. 19–24.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to information retrieval*. Cambridge University Press.

Mao, Junhua, Jiajing Xu, Kevin Jing, and Alan L. Yuille (2016). "Training and Evaluating Multimodal Word Embeddings with Large-scale Web Annotated Images". In: *NIPS*, pp. 442–450.

Mármol, Félix Gómez and Gregorio Martínez Pérez (2009). "Security threats scenarios in trust and reputation models for distributed systems". In: *Computers & Security* 28.7, pp. 545–556.

Marsh, Stephen Paul (1994). "Formalising Trust as a Computational Concept". PhD thesis. University of Stirling.

Marsh, Stephen and Pamela Briggs (2009). "Examining Trust, Forgiveness and Regret as Computational Concepts". In: *Computing with Social Trust*. Human-Computer Interaction Series. Springer, pp. 9–43.

Martens, James and Ilya Sutskever (2012). "Training Deep and Recurrent Networks with Hessian-Free Optimization". In: *Neural Networks: Tricks of the Trade (2nd ed.)* Vol. 7700. LNCS. Springer, pp. 479–535.

Massa, Paolo and Paolo Avesani (2007). "Trust-aware recommender systems". In: *RecSys*. ACM, pp. 17–24.

Mayer, Roger C, James H Davis, and F David Schoorman (1995). "An integrative model of organizational trust". In: *Academy of management review* 20.3, pp. 709–734.

McCabe, Kevin A, Mary L Rigdon, and Vernon L Smith (2003). "Positive reciprocity and intentions in trust games". In: *Journal of Economic Behavior & Organization* 52.2, pp. 267–275.

McCulloch, Warren S and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133.

McGlohon, Mary, Leman Akoglu, and Christos Faloutsos (2011). "Statistical Properties of Social Networks". In: *Social Network Data Analytics*. Springer, pp. 17–42.

McKnight, D. Harrison and Norman L. Chervany (2001). "What Trust Means in E-Commerce Customer Relationships: An Interdisciplinary Conceptual Typology". In: *Int. J. Electronic Commerce* 6.2, pp. 35–59.

McLaughlin, G Harry (1969). "SMOG grading: A new readability formula". In: *Journal of reading* 12.8, pp. 639–646.

Meo, Pasquale De, Katarzyna Musial-Gabrys, Domenico Rosaci, Giuseppe M. L. Sarnè, and Lora Aroyo (2017). "Using Centrality Measures to Predict Helpfulness-Based Reputation in Trust Networks". In: *ACM Trans. Internet Technol.* 17.1, 8:1–8:20.

Mertz, Jon (2013). *In Collaboration We Trust*. URL: https://www.thindifference.com/2013/04/in-collaboration-we-trust/ (visited on 05/17/2017).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *NIPS*, pp. 3111–3119.

Minka, Thomas P (2003). "A comparison of numerical optimizers for logistic regression". In: *Unpublished draft*.

Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar (2012). *Foundations of machine learning*. MIT press.

Molchanov, Dmitry, Arsenii Ashukha, and Dmitry Vetrov (2017). "Variational Dropout Sparsifies Deep Neural Networks". In: *CoRR* abs/1701.05369.

Monahan, Teresa, Gavin McArdle, and Michela Bertolotto (2008). "Virtual reality for collaborative e-learning". In: *Computers & Education* 50.4, pp. 1339–1353.

Morel, Nina J (2014). "Setting the Stage for Collaboration: An Essential Skill for Professional Growth." In: *Delta Kappa Gamma Bulletin* 81.1.

Morstatter, Fred, Jürgen Pfeffer, Huan Liu, and Kathleen M. Carley (2013). "Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose". In: *ICWSM*. The AAAI Press.

Mui, Lik (2002). "Computational models of trust and reputation: Agents, evolutionary games, and social networks". PhD thesis. Massachusetts Institute of Technology.

Mui, Lik, Mojdeh Mohtashemi, and Ari Halberstadt (2002). "A Computational Model of Trust and Reputation for E-businesses". In: *HICSS*. IEEE Computer Society, p. 188.

Murnighan, J Keith and Long Wang (2016). "The social world as an experimental game". In: *Organizational Behavior and Human Decision Processes* 136, pp. 80–94.

Nair, Vinod and Geoffrey E. Hinton (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *ICML*. Omnipress, pp. 807–814.

Natalie Kupferberg, MLS et al. (2011). "Accuracy and completeness of drug information in Wikipedia: an assessment". In: *Journal of the Medical Library Association* 99.4, p. 310.

N.D.Lewis (2016). *Deep Time Series Forecasting with Python*. CreateSpace Independent Publishing.

Nemoto, Keiichi, Peter A. Gloor, and Rob Laubacher (2011). "Social capital increases efficiency of collaboration among Wikipedia editors". In: *HT*. ACM, pp. 231–240.

Ng, Andrew (2013). *Machine Learning and AI via Brain simulations*.

Nguyen, David T. and John Canny (2007). "Multiview: improving trust in group video conferencing through spatial faithfulness". In: *CHI*. ACM, pp. 1465–1474.

Ning, Xia, Christian Desrosiers, and George Karypis (2015). "A Comprehensive Survey of Neighborhood-Based Recommendation Methods". In: *Recommender Systems Handbook*. Springer, pp. 37–76.

Nosek, John T. (1998). "The Case for Collaborative Programming". In: *Commun. ACM* 41.3, pp. 105–108.

Nov, Oded (2007). "What motivates Wikipedians?" In: *Commun. ACM* 50.11, pp. 60–64.

Ostrom, Elinor (2014). "Collective action and the evolution of social norms". In: *Journal of Natural Resources Policy Research* 6.4, pp. 235–252.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd (1999). *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford InfoLab.

Pan, Bing, Helene Hembrooke, Thorsten Joachims, Lori Lorigo, Geri Gay, and Laura A. Granka (2007). "In Google We Trust: Users' Decisions on Rank, Position, and Relevance". In: *J. Computer-Mediated Communication* 12.3, pp. 801–823.

Park, Do-Hyung, Jumin Lee, and Ingoo Han (2007). "The effect of on-line consumer reviews on consumer purchasing intention: The moderating role of involvement". In: *International journal of electronic commerce* 11.4, pp. 125–148.

Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio (2013). "On the difficulty of training recurrent neural networks". In: *ICML (3)*. Vol. 28. JMLR Workshop and Conference Proceedings. JMLR.org, pp. 1310–1318.

Pecori, Riccardo (2016). "S-Kademlia: A trust and reputation method to mitigate a Sybil attack in Kademlia". In: *Computer Networks* 94, pp. 205–218.

Pentina, Iryna and David G Taylor (2010). "Exploring source effects for online sales outcomes: the role of avatar-buyer similarity". In: *Journal of Customer Behaviour* 9.2, pp. 135–150.

Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena (2014). "DeepWalk: online learning of social representations". In: *KDD*. ACM, pp. 701–710.

Persson, Olle, Wolfgang Glänzel, and Rickard Danell (2004). "Inflationary bibliometric values: The role of scientific collaboration and the need for relative indicators in evaluative studies". In: *Scientometrics* 60.3, pp. 421–432.

Peters, Linda M and Charles C Manz (2007). "Identifying antecedents of virtual team collaboration". In: *Team Performance Management: An International Journal* 13.3/4, pp. 117–129.

Pineda, Fernando J (1987). "Generalization of back-propagation to recurrent neural networks". In: *Physical review letters* 59.19, p. 2229.

Pinsker, Joe (2015). *The Covert World of People Trying to Edit Wikipedia—for Pay*. URL: https://www.theatlantic.com/business/archive/2015/08/wikipedia-editors-for-pay/393926/ (visited on 06/02/2017).

Pinyol, Isaac and Jordi Sabater-Mir (2013). "Computational trust and reputation models for open multi-agent systems: a review". In: *Artif. Intell. Rev.* 40.1, pp. 1–25.

Poldrack, Russell (2017). "Neuroscience: The risks of reading the brain". In: *Nature* 541.7636, pp. 156–156.

Potthast, Martin, Benno Stein, and Robert Gerling (2008). "Automatic Vandalism Detection in Wikipedia". In: *ECIR*. Vol. 4956. Lecture Notes in Computer Science. Springer, pp. 663–668.

Pruitt, Dean G and Melvin J Kimmel (1977). "Twenty years of experimental gaming: Critique, synthesis, and suggestions for the future". In: *Annual review of psychology* 28.1, pp. 363–392.

Rabanal, Jean Paul and Daniel Friedman (2015). "How moral codes evolve in a trust game". In: *Games* 6.2, pp. 150–160.

Rapoport, Anatol (1973). *Two-person game theory*. Courier Corporation.

Resnick, Paul, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman (2000). "Reputation systems". In: *Communications of the ACM* 43.12, pp. 45–48.

Resnick, Paul, Richard Zeckhauser, John Swanson, and Kate Lockwood (2006). "The value of reputation on eBay: A controlled experiment". In: *Experimental economics* 9.2, pp. 79–101.

Ribeiro, Bruno F. and Donald F. Towsley (2010). "Estimating and sampling graphs with multidimensional random walks". In: *Internet Measurement Conference*. ACM, pp. 390–403.

Ribeiro, Marco Túlio, Sameer Singh, and Carlos Guestrin (2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *KDD*. ACM, pp. 1135–1144.

Richardson, Matthew and Pedro M. Domingos (2003). "Building large knowledge bases by mass collaboration". In: *K-CAP*. ACM, pp. 129–137.

Riegelsberger, Jens, M Angela Sasse, and John D McCarthy (2003). "The researcher's dilemma: evaluating trust in computer-mediated communication". In: *International Journal of Human-Computer Studies* 58.6, pp. 759–781.

Riegelsberger, Jens, Martina Angela Sasse, and John D. McCarthy (2005). "The mechanics of trust: A framework for research and design". In: *Int. J. Hum.-Comput. Stud.* 62.3, pp. 381–422.

Ristoski, Petar and Heiko Paulheim (2016). "RDF2Vec: RDF Graph Embeddings for Data Mining". In: *International Semantic Web Conference (1)*. Vol. 9981. Lecture Notes in Computer Science, pp. 498–514.

Rosaci, Domenico, Giuseppe M. L. Sarnè, and Salvatore Garruzzo (2012). "Integrating trust measures in multiagent systems". In: *Int. J. Intell. Syst.* 27.1, pp. 1–15.

Rosenblueth, Arturo and Norbert Wiener (1945). "The role of models in science". In: *Philosophy of science* 12.4, pp. 316–321.

Rotter, Julian B (1967). "A new scale for the measurement of interpersonal trust". In: *Journal of personality* 35.4, pp. 651–665.

Rousseau, Denise M, Sim B Sitkin, Ronald S Burt, and Colin Camerer (1998). "Not so different after all: A cross-discipline view of trust". In: *Academy of management review* 23.3, pp. 393–404.

Rowlands, Ian, David Nicholas, Peter Williams, Paul Huntington, Maggie Fieldhouse, Barrie Gunter, Richard Withey, Hamid R. Jamali, Tom Dobrowolski, and Carol Tenopir (2008). "The Google generation: the information behaviour of the researcher of the future". In: *Aslib Proceedings* 60.4, pp. 290–310.

Rozenshtein, Polina, Nikolaj Tatti, and Aristides Gionis (2017). "Finding Dynamic Dense Subgraphs". In: *ACM Trans. Knowl. Discov. Data* 11.3, 27:1–27:30.

Ruan, Yefeng and Arjan Durresi (2016). "A survey of trust management systems for online social communities–Trust modeling, trust inference and attacks". In: *Knowledge-Based Systems* 106, pp. 150–163.

Rumerhart, DE, GE Hinton, and RJ Williams (1986). "Learning representations by back-propagation errors". In: *Nature* 323, pp. 533–536.

Sally, Wehmeier et al. (2015). *Oxford Advanced Learner's Dictionary*.

Sapienza, Paola, Anna Toldra-Simats, and Luigi Zingales (2013). "Understanding trust". In: *The Economic Journal* 123.573, pp. 1313–1332.

Schmidhuber, Jürgen (2015). "Deep learning in neural networks: An overview". In: *Neural Networks* 61, pp. 85–117.

Science, Digital, John Hammersley, Ian Calvert, and Daniel Hook (2017). "The Connected Culture of Collaboration Report". In:

Senter, RJ and EA Smith (1967). *Automated readability index*. Tech. rep. DTIC Document.

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

ShareLatex (2017). *You're Part Of A Community Of More Than 1 Million*. URL: https://www.sharelatex.com/blog/2017/02/13/one-million-latex-users.html (visited on 05/15/2017).

Sherchan, Wanita, Surya Nepal, and Cécile Paris (2013). "A survey of trust in social networks". In: *ACM Comput. Surv.* 45.4, p. 47.

Siangliulue, Pao, Joel Chan, Kenneth C Arnold, Bernd Huber, Steven P Dow, and Krzysztof Z Gajos (2016). "Large-Scale Collaborative Innovation: Challenges, Visions and Approaches". In: *2016 AAAI Spring Symposium Series*.

Slonim, Robert and Ellen Garbarino (2008). "Increases in trust and altruism from partner selection: Experimental evidence". In: *Exp. Econ.* 11.2, pp. 134–153.

Slonim, Robert and Pablo Guillen (2010). "Gender selection discrimination: Evidence from a trust game". In: *Journal of Economic Behavior & Organization* 76.2, pp. 385–405.

Song, Dongjin and David A. Meyer (2015). "Link sign prediction and ranking in signed directed social networks". In: *Social Netw. Analys. Mining* 5.1, 52:1–52:14.

Sonnenwald, Diane H (2007). "Scientific collaboration". In: *Annual review of information science and technology* 41.1, pp. 643–681.

Sparx (2017). *Enterprise Architect: Distributed Teams and Collaboration*. URL: http://www.sparxsystems.com/enterprise-architect/distributed-teams-collaboration/distributed-teams-collaboration.html (visited on 05/15/2017).

Stanczyk, Urszula (2015). "Feature Evaluation by Filter, Wrapper, and Embedded Approaches". In: *Feature Selection for Data and Pattern Recognition*. Vol. 584. Studies in Computational Intelligence. Springer, pp. 29–44.

Stanczyk, Urszula and Lakhmi C. Jain (2015). "Feature Selection for Data and Pattern Recognition: An Introduction". In: *Feature Selection for Data and Pattern Recognition*. Vol. 584. Studies in Computational Intelligence. Springer, pp. 1–7.

Star, Susan Leigh and Karen Ruhleder (1994). "Steps Towards an Ecology of Infrastructure: Complex Problems in Design and Access for Large-Scale Collaborative Systems". In: *CSCW*. ACM, pp. 253–264.

Stein, Klaus and Claudia Hess (2007). "Does it matter who contributes: a study on featured articles in the german wikipedia". In: *Hypertext*. ACM, pp. 171–174.

Stvilia, Besiki, Michael B. Twidale, Linda C. Smith, and Les Gasser (2008). "Information quality work organization in Wikipedia". In: *JASIST* 59.6, pp. 983–1001.

Sun, Po-Ling and Cheng-Yuan Ku (2014). "Review of threats on trust and reputation models". In: *Industrial Management and Data Systems* 114.3, pp. 472–483.

Sundermeyer, Martin, Hermann Ney, and Ralf Schlüter (2015). "From Feedforward to Recurrent LSTM Neural Networks for Language Modeling". In: *IEEE/ACM Trans. Audio, Speech & Language Processing* 23.3, pp. 517–529.

Suzuki, Yu (2015). "Quality Assessment of Wikipedia Articles Using h-index". In: *Journal of Information Processing* 23.1, pp. 22–30.

Suzuki, Yu and Satoshi Nakamura (2016). "Assessing the Quality of Wikipedia Editors through Crowdsourcing". In: *WWW*. ACM, pp. 1001–1006.

Sztompka, Piotr (1999). *Trust: A sociological theory*. Cambridge University Press.

Tadelis, Steven (2013). *Game theory: an introduction*. Princeton University Press.

Takemoto, Kazuhiro and Chikoo Oosawa (2012). "Introduction to complex networks: measures, statistical properties, and models". In: *Statistical and Machine Learning Approaches for Network Analysis*. Hoboken, New Jersey: John Wiley & Sons, Inc, pp. 45–75.

Tan, Xin and Yongbeom Kim (2015). "User acceptance of SaaS-based collaboration tools: a case of Google Docs". In: *Journal of Enterprise Information Management* 28.3, pp. 423–442.

Tang, Jiliang, Shiyu Chang, Charu C. Aggarwal, and Huan Liu (2015). "Negative Link Prediction in Social Media". In: *WSDM*. ACM, pp. 87–96.

Tang, Jiliang, Yi Chang, Charu Aggarwal, and Huan Liu (2016). "A Survey of Signed Network Mining in Social Media". In: *ACM Comput. Surv.* 49.3, 42:1–42:37.

Tang, Jiliang, Huiji Gao, Huan Liu, and Atish Das Sarma (2012). "eTrust: understanding trust evolution in an online world". In: *KDD*. ACM, pp. 253–261.

Tang, Jiliang, Xia Hu, and Huan Liu (2013). "Social recommendation: a review". In: *Social Netw. Analys. Mining* 3.4, pp. 1113–1133.

Tang, John C and Scott L Minneman (1991). "VideoDraw: a video interface for collaborative drawing". In: *ACM Transactions on Information Systems (TOIS)* 9.2, pp. 170–184.

Tavakolifard, Mozhgan and Kevin C. Almeroth (2012). "A Taxonomy to Express Open Challenges in Trust and Reputation Systems". In: *JCM* 7.7, pp. 538–551.

Teacy, W. T. Luke, Jigar Patel, Nicholas R. Jennings, and Michael Luck (2006). "TRAVOS: Trust and Reputation in the Context of Inaccurate Information Sources". In: *Autonomous Agents and Multi-Agent Systems* 12.2, pp. 183–198.

Thung, Ferdian, Tegawendé F. Bissyandé, David Lo, and Lingxiao Jiang (2013). "Network Structure of Social Coding in GitHub". In: *CSMR*. IEEE Computer Society, pp. 323–326.

Thurstone, Louis L (1928). "Attitudes can be measured". In: *American journal of sociology* 33.4, pp. 529–554.

Tomasello, Michael, Alicia P Melis, Claudio Tennie, Emily Wyman, and Esther Herrmann (2012). "Two key steps in the evolution of human cooperation". In: *Current Anthropology* 53.6, pp. 673–692.

Tramullas, Jesús, Piedad Garrido Picazo, and Ana I. Sánchez-Casabón (2016). "Research on Wikipedia Vandalism: a brief literature review". In: *CERI*. ACM, p. 15.

Trottier, Daniel (2016). *Social media as surveillance: Rethinking visibility in a converging world*. Routledge.

Tucker, Albert (1950). *A two person dilemma*. Lecture at Stanford University.

Ugander, Johan, Brian Karrer, Lars Backstrom, and Cameron Marlow (2011). "The Anatomy of the Facebook Social Graph". In: *CoRR* abs/1111.4503.

Vaca, Rodrigo (2015). *Celebrating 10 Years: Zoho CRM Free For SMBs with Up To 10 Users*. URL: https://www.zoho.com/crm/blog/celebrating-10-years-zoho-crm-free-for-smbs-with-up-to-10-users.html (visited on 04/21/2017).

Vanberg, Christoph (2008). "Why do people keep their promises? an experimental test of two explanations". In: *Econometrica* 76.6, pp. 1467–1480.

Vishwanathan, S. V. N., Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt (2010). "Graph Kernels". In: *Journal of Machine Learning Research* 11, pp. 1201–1242.

Vu, Quang Hieu, Mihai Lupu, and Beng Chin Ooi (2010). "Trust and reputation". In: *Peer-to-Peer Computing*. Springer, pp. 183–214.

Wade, Nicholas (2011). *Supremacy of a Social Network*. URL: http://www.nytimes.com/2011/03/15/science/15humans.html (visited on 06/02/2017).

Wales, Jimmy and Larry Sanger (2001). *Wikipedia: The free encyclopedia*. URL: https://en.wikipedia.org/wiki/Wikipedia (visited on 04/21/2017).

Wang, Guan-Nan, Hui Gao, Lian Chen, Dennis NA Mensah, and Yan Fu (2015). "Predicting positive and negative relationships in large social networks". In: *PloS one* 10.6, e0129530.

Wang, Haohan, Bhiksha Raj, and Eric P. Xing (2017). "On the Origin of Deep Learning". In: *CoRR* abs/1702.07800.

Wang, Jing, Jie Shen, Ping Li, and Huan Xu (2017). "Online Matrix Completion for Signed Link Prediction". In: *WSDM*. ACM, pp. 475–484.

Wang, Suhang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu (2017). "Signed network embedding in social media". In: *SDM*. SIAM, pp. 327–335.

Wang, Yao and Julita Vassileva (2007). "A Review on Trust and Reputation for Web Service Selection". In: *ICDCS Workshops*. IEEE Computer Society, p. 25.

Warncke-Wang, Morten, Vladislav R. Ayukaev, Brent Hecht, and Loren G. Terveen (2015). "The Success and Failure of Quality Improvement Projects in Peer Production Communities". In: *CSCW*. ACM, pp. 743–756.

Warncke-Wang, Morten, Dan Cosley, and John Riedl (2013). "Tell me more: an actionable quality model for Wikipedia". In: *OpenSym*. ACM, 8:1–8:10.

Weisberg, Jacob, Dov Te'eni, and Limor Arman (2011). "Past purchase and intention to purchase in e-commerce: The mediation of social presence and trust". In: *Internet research* 21.1, pp. 82–96.

Wikimedia (2016). *Objective Revision Evaluation Service*. URL: https://meta.wikimedia.org/wiki/Objective_Revision_Evaluation_Service/wp10 (visited on 04/24/2017).

— (2017). *Wikipedia Datasets*. URL: https://meta.wikimedia.org/wiki/Datasets (visited on 06/10/2017).

Wikipedia (2017a). *Edit count*. URL: https://en.wikipedia.org/wiki/Wikipedia:Edit_count (visited on 06/12/2017).

— (2017b). *Grading scheme*. URL: https://en.wikipedia.org/wiki/Template:Grading_scheme (visited on 06/12/2017).

— (2017c). *User contributions*. URL: https://en.wikipedia.org/wiki/Help:User_contributions (visited on 06/12/2017).

— (2017d). *Vandalism on Wikipedia*. URL: https://en.wikipedia.org/wiki/Vandalism_on_Wikipedia (visited on 06/10/2017).

— (2017e). *Wiki*. URL: https://en.wikipedia.org/wiki/Wiki (visited on 07/10/2017).

— (2017f). *Wikipedia:Edit count*. URL: https://en.wikipedia.org/wiki/Wikipedia:Edit_count (visited on 05/02/2017).

— (2017g). *WikiProject Wikipedia/Assessment*. URL: https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Wikipedia/Assessment (visited on 06/09/2017).

WikiTrust (2017). *A reputation system for Wikipedia authors and content*. URL: http://wikitrust.soe.ucsc.edu (visited on 04/24/2017).

Wilde, Louis L (1981). "On the use of laboratory experiments in economics". In: *Philosophy in Economics*. Springer, pp. 137–148.

Wilkinson, Dennis M. and Bernardo A. Huberman (2007). "Cooperation and quality in Wikipedia". In: *Int. Sym. Wikis*. ACM, pp. 157–164.

Wilson, Rick K and Catherine C Eckel (2006). "Judging a book by its cover: Beauty and expectations in the trust game". In: *Political Research Quarterly* 59.2, pp. 189–202.

Winer, Ben James, Donald R Brown, and Kenneth M Michels (1971). *Statistical principles in experimental design*. Vol. 2. McGraw-Hill New York.

Wöhner, Thomas and Ralf Peters (2009). "Assessing the quality of Wikipedia articles with lifecycle based metrics". In: *Int. Sym. Wikis*. ACM.

Wood, Michael (2011). "Collaborative lab reports with Google Docs". In: *The Physics Teacher* 49.3, pp. 158–159.

Wu, Guangyu, Martin Harrigan, and Pádraig Cunningham (2012). "Classifying Wikipedia articles using network motif counts and ratios". In: *WikiSym*. ACM, p. 12.

Wu, Zhaoming, Charu C. Aggarwal, and Jimeng Sun (2016). "The Troll-Trust Model for Ranking in Signed Networks". In: *WSDM*. ACM, pp. 447–456.

Xiong, Li and Ling Liu (2004). "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities". In: *IEEE Trans. Knowl. Data Eng.* 16.7, pp. 843–857.

Xu, Yanxiang and Tiejian Luo (2011). "Measuring article quality in Wikipedia: Lexical clue model". In: *Proc. of SWS*, pp. 141–146.

Yamagishi, Toshio, Satoshi Akutsu, Kisuk Cho, Yumi Inoue, Yang Li, and Yoshie Matsumoto (2015). "Two-Component Model of General Trust: Predicting Behavioral Trust from Attitudinal Trust". In: *Social Cognition* 33.5, p. 436.

Yamamoto, Atsushi, Daisuke Asahara, Tomoko Itao, Satoshi Tanaka, and Tatsuya Suda (2004). "Distributed Pagerank: A Distributed Reputation Model for Open Peer-to-Peer Network". In: *SAINT Workshops*. IEEE Computer Society, pp. 389–394.

Yao, Xin and Paul J Darwen (1999). "How important is your reputation in a multi-agent environment". In: *SMC*. Vol. 2. IEEE, pp. 575–580.

Ye, Jihang, Hong Cheng, Zhe Zhu, and Minghua Chen (2013). "Predicting positive and negative links in signed social networks by transfer learning". In: *WWW*. ACM, pp. 1477–1488.

Yen, Steven T (2002). "An econometric analysis of household donations in the USA". In: *Applied Econ. Letters* 9.13, pp. 837–841.

Yim, Soobin, Dakuo Wang, Judith S. Olson, Viet Vu, and Mark Warschauer (2017). "Synchronous Collaborative Writing in the Classroom: Undergraduates' Collaboration Practices and their Impact on Writing Style, Quality, and Quantity". In: *CSCW*. ACM, pp. 468–479.

You, Qiang, Ou Wu, Guan Luo, and Weiming Hu (2016). "A Probabilistic Matrix Factorization Method for Link Sign Prediction in Social Networks". In: *MLDM*.

Yuan, Shuhan, Xintao Wu, and Yang Xiang (2017). "SNE: Signed Network Embedding". In: *PAKDD (2)*. Vol. 10235. Lecture Notes in Computer Science, pp. 183–195.

Yuksel, Beste F, Penny Collisson, and Mary Czerwinski (2017). "Brains or Beauty: How to Engender Trust in User-Agent Interactions". In: *ACM Transactions on Internet Technology (TOIT)* 17.1, p. 2.

Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals (2015). "Recurrent Neural Network Regularization". In: *CoRR* abs/1409.2329.

Zha, Yilong, Tao Zhou, and Changsong Zhou (2016). "Unfolding large-scale online collaborative human dynamics". In: *Proceedings of the National Academy of Sciences* 113.51, pp. 14627–14632.

Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals (2017). "Understanding deep learning requires rethinking generalization". In: *ICLR*.

Zhang, Nelson (2010). "Running the turk: interview with Amazon.com vice president Sharon Chiarella and PR manager Kay Kinton". In: *ACM Crossroads* 17.2, pp. 50–51.

Zhang, Richong and Yongyi Mao (2014). "Trust Prediction via Belief Propagation". In: *ACM Trans. Inf. Syst.* 32.3, 15:1–15:27.

Zheng, Jun, Nathan Bos, Judith S. Olson, and Gary M. Olson (2001). "Trust with*out* touch: jump-start trust with social chat". In: *CHI Extended Abstracts*. ACM, pp. 293–294.

Zheng, Xiaolong, Daniel Dajun Zeng, and Fei-Yue Wang (2015). "Social balance in signed networks". In: *Information Systems Frontiers* 17.5, pp. 1077–1095.

Zheng, Yudian, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng (2017). "Truth Inference in Crowdsourcing: Is the Problem Solved". In: *VLDB*, pp. 541–552.

Zhou, Jiufeng, Lixin Han, Yuan Yao, Xiaoqin Zeng, and Feng Xu (2014). "A Parallel Approach to Link Sign Prediction in Large-Scale Online Social Networks". In: *Comput. J.*

Appendix A

Background Knowledge

Contents

In this chapter, we describe some fundamentals knowledge which will be used in the following chapters.

A.1 Trust Game

A.1.1 Game design

Trust game is one of behavioral games that encourage participants not only to compete but also to cooperate with other participants [Murnighan and L. Wang, 2016]. Trust game is considered as an extended variation of the classical prisoner's dilemma [Kendall et al., 2007] in game theory. The standard trust game design is presented by [Berg et al., 1995][27].

Trust game is visualized in Figure A.1. In the most simple form [Berg et al., 1995], trust game contains two participants or two players, one is called *sender*[28] and the other one is called *receiver*[29]. The game is played by turn. The sender plays first by selecting an amount of money between 0 and 10 to sends to the receiver. The money will be tripled before the receiver receives it. Then the receiver can play by selecting an amount of money between 0 and what he received to send back to the sender. This time, the sending amount will not be tripled but kept the same.

[27] In fact, Berg called their game as *investment game*, but many follow-up studies used the term *trust game* [Johnson and Mislin, 2011; Murnighan and L. Wang, 2016; Cooper and Kagel, 2016], while for several other research works the term *trust game* is used to refer sequential prisoner dilemma setting [Riegelsberger, M Angela Sasse, et al., 2003; Rabanal and Friedman, 2015]. To be consistent, in this thesis, we used *trust game* to refer the game presented by [Berg et al., 1995], and *sequential prisoner dilemma* for the other game.

[28] In literature, this player is also called *trustor* [Engle-Warnick and Robert L. Slonim, 2004] or *first mover* [McCabe et al., 2003] or just simply *player A* [Fehr et al., 2003].

[29] Similarly, this player is called *trustee*, *second mover*, or *player B* also.

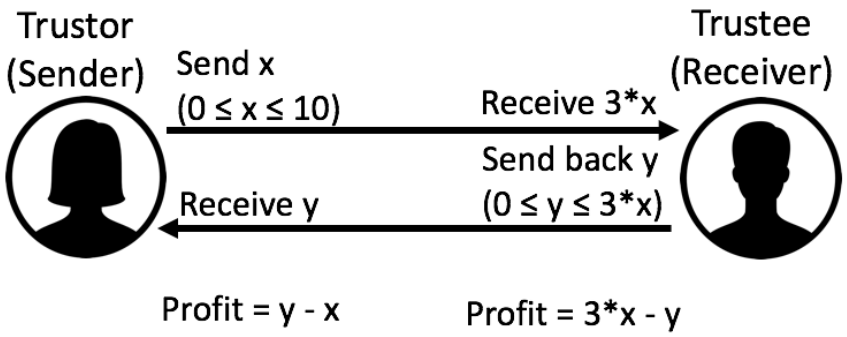


Figure A.1: Trust game

The goal of each participant is to maximize their own profit.

The exchanging amounts between players are considered as representation of trust between them [Glaeser et al., 2000; Fehr et al., 2003; Bellemare and Kröger, 2007; Brülhart and Usunier, 2012; Sapienza et al., 2013].

Let's give an example with two players: Alice as the sender and Bob as the receiver. In the first move, Alice sends 5 to Bob. Therefore, Bob will receive 15 to his balance. Then Bob sends back 7 to Alice. This time, Alice receives 7 only. In the end, the profit of this turn for Alice and Bob are 7 and 8 respectively.

Based on the definition of collaboration discussed in Chapter 1, trust game is indeed a collaborative environment because:

- Participants need to collaborate to earn a higher reward value for everyone.
- However, a malicious participant can deviate to gain their own profit while harming other participant.

A.1.2 Game analysis

One of the basic assumptions in economics and game theory is that the participants are self-interests and rational [Camerer, 2003], means that:

- A participant has one and only one target which is to maximize their own profit. The participant is not bounded by any ethical rule but only the rules of the game.
- A participant has infinite computational power. It means, given a scenario with complete or incomplete information, the participant can always reason and make the optimized decision.
- A participant knows that other participants share the same two above characteristics.

For self-interested subjects, a subgame perfect Nash equilibrium [Tadelis, 2013, Chapter 8] predicts that both players should send nothing [Camerer, 2003; Murnighan and L. Wang, 2016].

The reasoning process is as follows. Any other sending back amount y that $y \geq 0$ will reduce the receiver's profit. According to theory, the sender knows this fact, so at her turn, she should send 0 to the receiver because she knows that she will not receive anything back regardless what she sends [Camerer, 2003]. It is noted that the analysis only holds for one-shot trust game. To

our knowledge, the theoretical analysis for users behavior in repeated trust game is still an open question [Bruttel and Kamecke, 2012; Breitmoser, 2015].

Despite the theoretical analysis, the zero-behavior when both participants send 0 is not dominant in large scale experiments. In fact, the sending and sending back amounts follow the normal distribution [Berg et al., 1995; Johnson and Mislin, 2011; Cooper and Kagel, 2016; Murnighan and L. Wang, 2016].

The game has been adapted using a vast range of different conditions [Chakravarty et al., 2011; Johnson and Mislin, 2011]. The original game was single-trial, i.e. consisting of a single interaction between two users, specifically to isolate trust from reputation. However, one interaction is not enough for building a trust relationship between users. Therefore the game has been extended with repeated trials [Cochard et al., 2004; Engle-Warnick and Robert L Slonim, 2006; Engle-Warnick and Robert L Slonim, 2001]. The length of the repeated trust game could be undefined [Engle-Warnick and Robert L. Slonim, 2006] or fixed [Dubois et al., 2012], which may affect participant strategies. Different studies employ different conditions, mostly concerning the provision of partner information. The authors of [R. Slonim and Garbarino, 2008] provided partner gender, age and income information. The historical log of partners' action can also be provided to players [Berg et al., 1995; Bracht and Feltovich, 2009; Gary E. Bolton et al., 2005; Dubois et al., 2012]. Some studies analyze players behavior when they are allowed to communicate during the game [Vanberg, 2008; Bracht and Feltovich, 2009]. Other research studies simulate business *contracts* to allow players to setup contracts between each other [Avner Ben-Ner and Putterman, 2009; Braynov and Sandholm, 2002; Colombo and Merzoni, 2006; Feltovich and Swierzbinski, 2011] or provide pre-commitments prior to the game start [Bracht and Feltovich, 2008]. There are several papers that use the trust game to study trust transfer, i.e. trust between contexts [Buntain and Golbeck, 2015; Delgado-Márquez et al., 2012].

Predicting users' behavior before the trust game starts is the subject of numerous studies. All of the studies we are aware of in this topic focused on the one-trial trust game. [Gunnthorsdottir et al., 2002] used the Mach test to determine participant personality characteristics before the game experiment and then used the resulting Mach score to predict participant game behavior. [Evans and Revelle, 2008] predicted users' behavior based on previously collected responses to the *Propensity to Trust Survey*. Using a similar idea, [Yamagishi et al., 2015] defined *attitudinal trust* calculated from prior questionnaire responses, to predict user behavior. [Yen, 2002] claimed that participants with higher income send more to their partners than users with lower income. [Falk, Meier, et al., 2010] confirmed this suggestion by showing that students tend to send less than other social groups.

[Yao and Darwen, 1999] showed that displaying reputation score can increase the cooperation rate in repeated prisoner dilemma game which can be considered as a simple version of repeated trust game.

[Bente, Dratsch, Rehbach, et al., 2014] tested the influence of avatar and reputation levels on buyers' decisions. The authors showed that reputation score and avatars could encourage the buying decision of buyers. However, the authors did not study the behavior of sellers (receivers in our paper), and the reputation scores are artificial rather than computing from real behavior. Different from this work, we studied the effect of trust score which is computed based on real behavior of participants in the experiment. [Lunawat, 2013] studied the building process of reputation in repeated trust game while the receiver can decide to disclose or not her private information. In the studied game participants depend on their partners decision on providing information. In our approach, each participant can calculate trust score of any partner based on their observed behavior without any dependency. [Yuksel et al., 2017] studies an interesting research question: which is more important to build trust, reliability or attractiveness? The

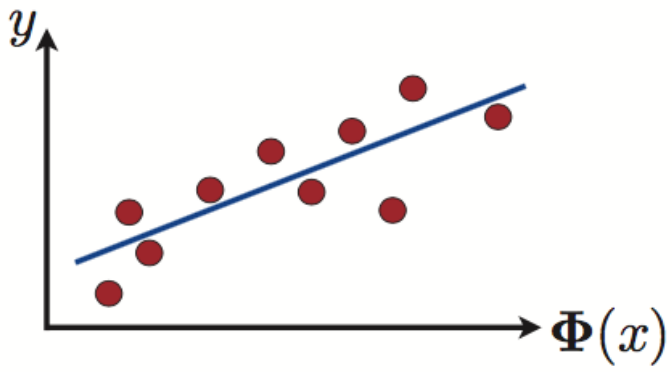


Figure A.2: Linear regression

study indeed suggested that, having a beautiful avatar is more important.

Trust game has been used to test the effect of different mechanisms in human trust [Riegelsberger, Martina Angela Sasse, et al., 2005]. However, to the best of our knowledge there is not yet a study of the influence of trust score on human behavior.

A.2 Machine Learning Basics

In this section, we describe several machine learning and deep learning techniques that will be used in following chapters. We also describe how do we validate the algorithms, and the metrics to measure the performance of these algorithms.

A.2.1 Shallow machine learning

In fact, the term "shallow machine learning" is used only after the term "deep learning" was invented[30]. Today, the term "shallow machine learning" is used to refer to any algorithm before the *deep learning era*, includes linear and logistic regression, support vector machine (SVM), decision tree, and random forest.

A.2.1.1 Linear regression

Linear regression could be considered as the most simple machine learning algorithm. Given a set of feature $X_1, X_2, ...X_k$ with the corresponding output Y, the linear regression algorithm tries to estimate the following formula:

$$y = \sum_{i=1}^{k} \alpha_i * x_i + \beta \tag{A.1}$$

The value α_i and β are estimated by minimizing the difference between the estimated values and the real values of Y [Mohri et al., 2012, Chapter 10]. The difference metric which is usually used is *mean squared error*, defined as:

$$\frac{1}{k}\sum_{i=1}^{k}(\alpha_i * x_i + \beta - y_i)^2 \tag{A.2}$$

[30] http://deeplearning.net/

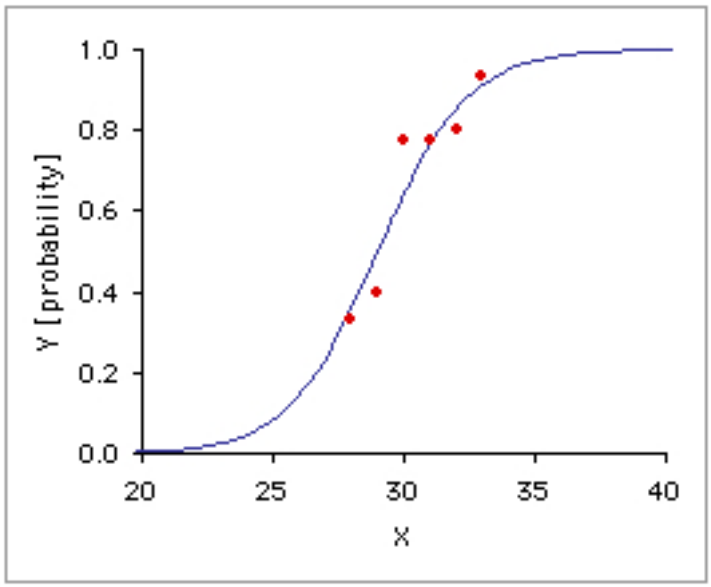


Figure A.3: Logistic regression

A.2.1.2 Logistic regression

Logistic regression [Lesmeister, 2015, Chapter 3] is visualized in Figure A.3. Logistic regression estimates the probability that $Y = 1$ as:

$$Pr(Y = 1) = \frac{1}{1 + e^{-\beta - \sum_{i=1}^{k} \alpha_i * x_i}} \tag{A.3}$$

Originally, logistic regression can work only in binary classification problem. However, by setting up multiple conditional probability, we can extend the logistic regression to multi-class classification problem, which is called *multinomial logistic regression* [Böhning, 1992].

A.2.1.3 Support vector machine (SVM)

The idea of SVM [Gollapudi, 2016, Chapter 6] is to build a *hyperplane* to separate two classes of objects, as visualized in Figure A.4. Different from logistic regression, the idea of SVM can be easily extended to multi-class case.

A.2.1.4 Decision Tree & Random Forest

Decision tree [Gollapudi, 2016, Chapter 5] is visualized in Figure A.5. Following a decision tree to classify is obvious: we start at a root of tree and just follow the guidance at each node until a leaf.

In practice, a decision tree is considered as a *weak* learning algorithm [Galar et al., 2012]. An updated version of decision tree, *random forest* is used more popular. The idea of *random forest* is to build many trees as visualized in Figure A.6 then used majority voting to decide the final output value.

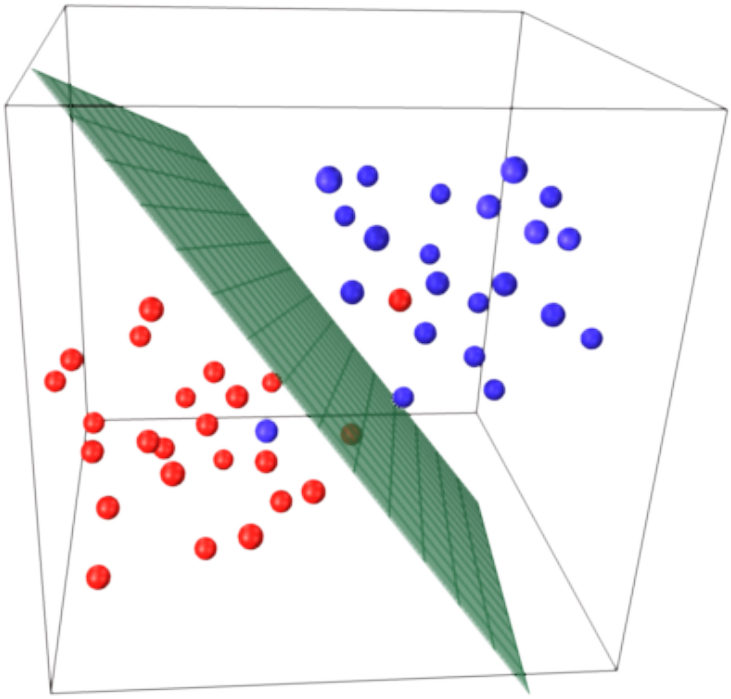


Figure A.4: Support vector machines

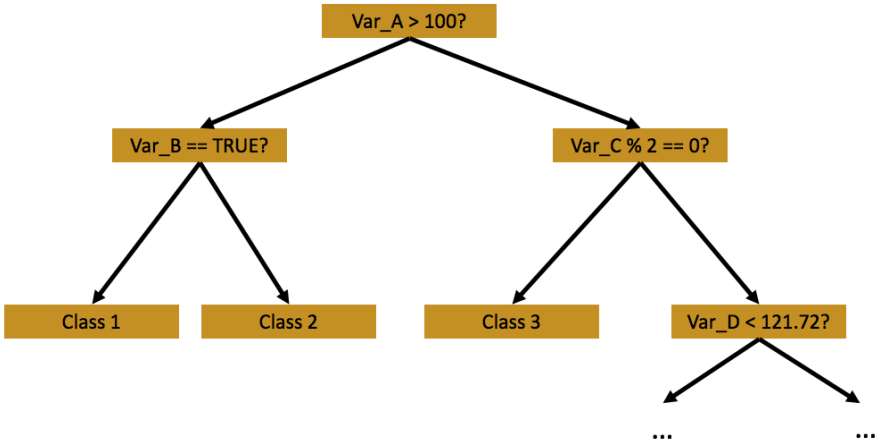


Figure A.5: Decision tree

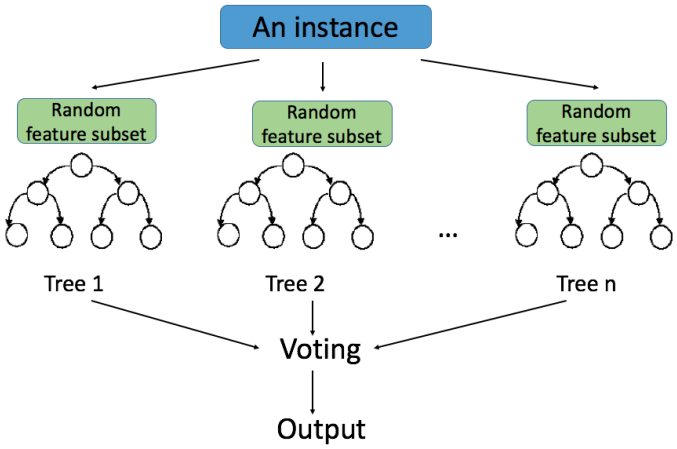


Figure A.6: Random forest

A.2.2 Deep learning

Deep learning is a part of machine learning which attempts to model high-level abstract of data [L. Deng and Yu, 2014]. Deep learning research attracts a lot of attention from both academy and industry. Deep learning techniques have been applied successfully in many topics, including image processing, video processing and natural language processing [Schmidhuber, 2015]. In fact, the deep learning theory has been introduced in 1980s [Rumerhart et al., 1986; Pineda, 1987], but in practice, the breakthrough appeared only in 2006 with the work of Hinton in digit recognition using Restricted Boltzmann Machine [Goodfellow et al., 2016].

In this section we presented some fundamental knowledge to understand and use deep learning techniques in our studies. Readers should refer to two intensive reviews [Schmidhuber, 2015; H. Wang et al., 2017] or the deep learning textbook [Goodfellow et al., 2016] for more details information.

A.2.2.1 Deep neural networks

Artificial neural network was invented in 1943 [McCulloch and Pitts, 1943] as a mathematical model of human brains. An artificial neural network includes at least one input and one output layer, and optional one or many hidden layers, and each layer includes one or many neurons. An artificial neural network with multiple hidden layers is called *deep neural network*, but so far there is not yet a standard definition of how many hidden layers as a minimum number for an artificial neural network to be considered as be *deep* [Schmidhuber, 2015]. Basically we can consider all the neural networks with at least two hidden layers as a deep neural network.

A visualization of a DNN is displayed in Figure A.7. The layer k computes an output vector h^{k} using the output $h^{\mathrm{k\text{-}1}}$ of the previous layer, starting with the input layer $x = h^0$ as in Equation A.4, with b^{k} is offset vector and W^{k} is matrix of weights.

$$h^{\mathrm{k}} = f(b^{\mathrm{k}} + W^{\mathrm{k}} h^{\mathrm{k\text{-}1}}) \tag{A.4}$$

The function f in Equation A.4 is called *activation function*, which decides how each neuron calculates and transfers the signal to the neurons in subsequent layer. The advantage of deep neural networks is that, the calculation in deep neural networks usually requires simple mathematical activation functions. A popular activation function is called *rectifier*, which is the most

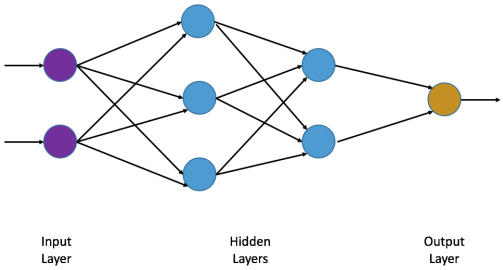


Figure A.7: A multi-layer feed-forward neural network.

simple non-linear function. Rectifier function is defined as:

$$rectifier(x) = max(0, x) \tag{A.5}$$

Even the *rectifier* function is very simple, it can produce a good performance in deep learning [Nair and Hinton, 2010]. The advantage of *rectifier* activation function in compare with other activation function like *tanh* or *sigmoid* is that it can speed up the training process [Goodfellow et al., 2016].

As we observe in Equation A.4, training and applying deep neural networks could be considered as a series of matrix calculation. These operations can be calculated very fast today using parallel computing techniques, such as Hadoop and MapReduce [Dean and Ghemawat, 2008] or GPU-computing technology likes CUDA [Chetlur et al., 2014].

Deep neural networks are usually trained by using gradient descent [Bengio, 2012]. The goal of training phase is to minimize the error loss on training data, i.e. to minimize the difference between the actual output values and the predicting output values of the DNN model on the training data [Martens and Sutskever, 2012]. The issue of gradient descent training is that it might stuck in local minima [Bengio, 2009]. The problem is solved practically by tuning the parameter number of training steps.

A.2.2.2 Recurrent neural networks

Recurrent neural networks (RNN) is a class of dynamic models which are used for sequence generation in different domains [Graves, 2013]. The connections between neurons in RNN can form cycles [Graves, 2012]. The visualization of a RNN is displayed in Figure A.8. The input of RNN is a token series $X = x_1, x_2, ..x_T$ ordered in time. The input will be passed through a stack of hidden layer to compute the output y. The output then will be used as a part of the next input sequence to predict the next input token. In other words we use the input x_t and the output y_t to predict the distribution of the input x_{t+1}.

The hidden layers are computed as:

$$h_t^1 = H(W_{ih^1}x_t + W_{h^1h^1}h_{t-1}^1 + b_h^1) \tag{A.6}$$

$$h_t^n = H(W_{ih^n}x_t + W_{h^{n-1}h^n}h_t^{n-1} + W_{h^nh^n}h_{t-1}^n + b_h^n) \tag{A.7}$$

wherein:

- h_t^n is the output of the n^{th} hidden layer at time t.

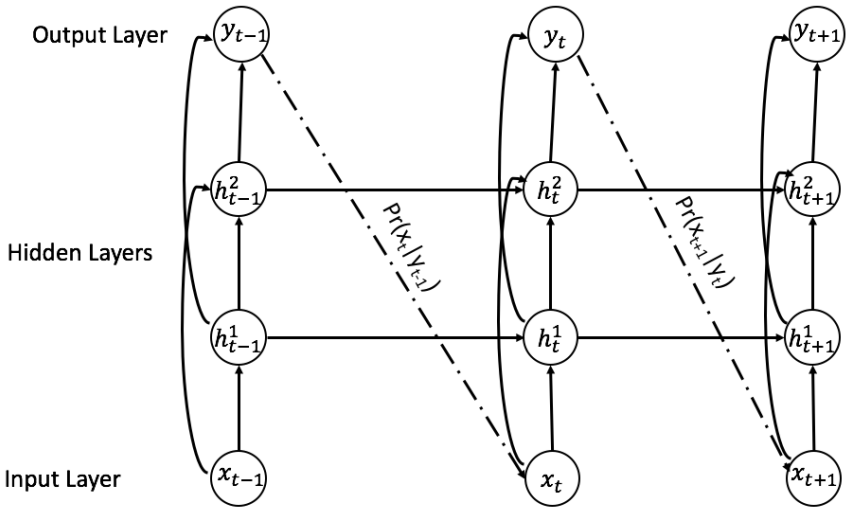


Figure A.8: Deep Recurrent Neural Network [Graves, 2013].

- b is the bias vector. b_h^n is the bias vector of the n^{th} hidden layer. b_y is the bias vector of the output layer.
- W is the weight matrices, with W_{ih^n} is the matrix between the input and the n^{th} hidden layer, and $W_{h^j h^k}$ is the matrix connects the j^{th} hidden layer and k^{th} hidden layer. If $j == k$ means that it is a recurrent connection.
- H is the activation function. In the past the *sigmoid* is usually used as the activation function [J. Chung et al., 2014] but today the LSTM cell is used more often [N.D.Lewis, 2016, Chapter 7,8].

RNN is claimed as a universal model [Goodfellow et al., 2016], means that it can compute any function computable by a Turing machine. If the model is large enough, RNN can gradually learn any sequence at any complexity level [Graves, 2013].

Up to now, one of the most effective sequence models is LSTM [Goodfellow et al., 2016]. The performance of LSTM have been proved empirically in several studies [Graves, 2013; J. Chung et al., 2014; Józefowicz et al., 2015]. The idea of LSTM is to replace the activation unit inside a RNN cell, which is traditional a *tanh* activation [J. Chung et al., 2014], by a LSTM unit as visualized in Figure A.9.

The output of a LSTM cell is calculated as:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{A.8}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{A.9}$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{A.10}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \tag{A.11}$$

$$h_t = o_t \tanh(c_t) \tag{A.12}$$

wherein: σ is the logistic sigmoid function; i, f, o are *input gate, forget gate* and *output gate* respectively, and c is *memory* cell.

After we computed the hidden sequences h_T, the output of RNN is computed as:

$$\hat{y}_t = b_y + \sum_{n=1}^{N} W_{h^n y} h_t^n \tag{A.13}$$

$$y_t = \gamma(\hat{y}_t) \tag{A.14}$$

where γ is the activation function of output layer.

The probability of the input sequence X is:

$$Pr(X) = \prod_{t=1}^{T} Pr(x_{t+1}|y_t) \tag{A.15}$$

and the loss function which is used to train the network is defined as:

$$L(X) = -\sum_{t=1}^{T} \log Pr(x_{t+1}|y_t) \tag{A.16}$$

In the last LSTM layer, we trained a softmax cost function to predict the quality class as:

$$\delta(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for } j = 1..K \tag{A.17}$$

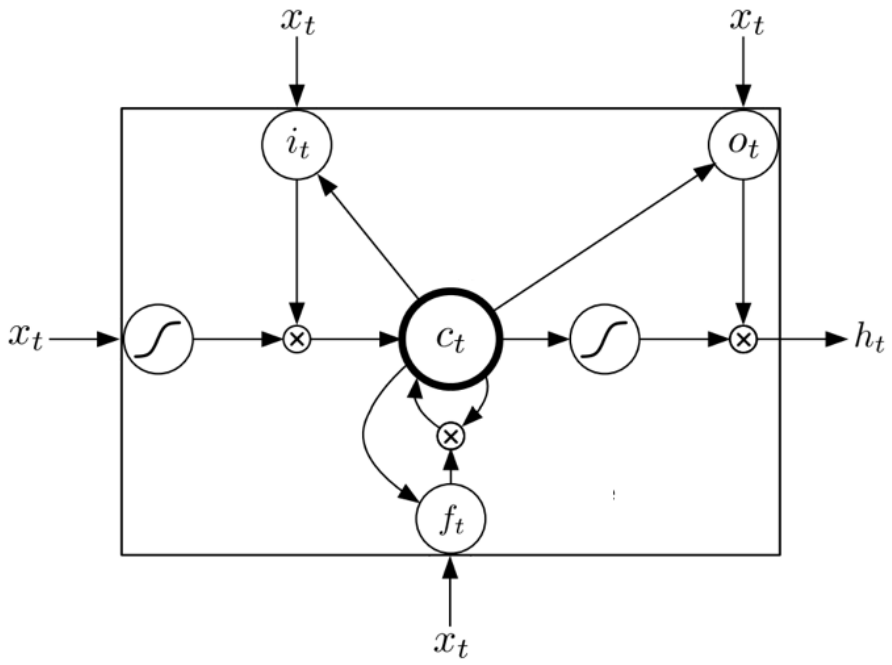


Figure A.9: Long-Short Term Memory Cell [Graves, 2013].

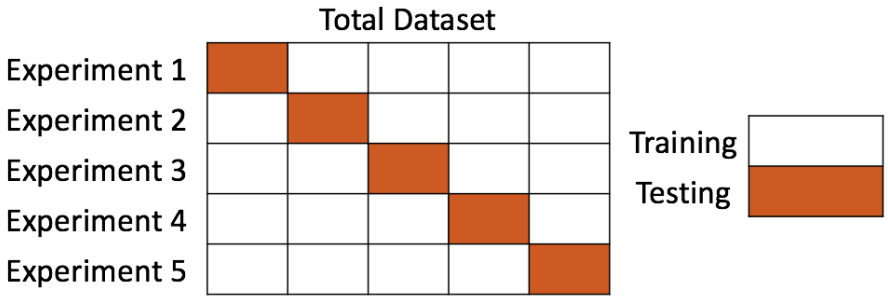


Figure A.10: 5-fold cross validation

Most machine learning algorithms are designed to work on the data where all instances have the same length [Shalev-Shwartz and Ben-David, 2014] while in our research studies the input data have different length between instances. However, by design, RNN is perfectly fit with varied-length data because it can roll back or forward with a flexible number of step. In practice, among other deep architectures, RNN usually be used in natural language processing field [Goodfellow et al., 2016; J. Mao et al., 2016], especially in language modelling [Grave et al., 2016; Gal and Ghahramani, 2016; Sundermeyer et al., 2015].

Originally, RNN was designed for time-series analysis [Connor et al., 1994], so it only learn from the historical data. However, RNN can be extended to bidirectional RNN [Graves, 2012; Graves, 2013]. The core idea of bidirectional RNN is to have two RNN to learn the data. One RNN will learn from the beginning of the data while the other learns from the end.

Stateful LSTM Originally if the training data is modified, we need to train RNN-LSTM from scratch again. However, using *stateful* LSTM model [Gers et al., 2000] we can continuously train the model on new data without retraining old data. The core idea of stateful LSTM is to remember the states of each batch [Pascanu et al., 2013] and use these states as the initial states of the next training batch [N.D.Lewis, 2016].

A.2.3 Validation & Metrics

A.2.3.1 Cross validation

In order to evaluate a predicting algorithm, a dataset will be single-divided into a training set and a testing set, such as 80% of the dataset is used for training and the remaining 20% of the dataset is used for testing [Warncke-Wang, Ayukaev, et al., 2015]. However, it could lead to bias in the evaluation result [James et al., 2013, Chapter 5]. Therefore, a *k-fold* cross-validation is used.

The idea of k-fold cross validation is, the dataset is divided into k equal parts. The algorithm which is evaluated will be performed k times. For each run, a single part will be used as the testing set, while the remaining $k-1$ parts will be used as the training set. In practice, the value of k is usually set to 5 or 10 [James et al., 2013, Chapter 5]. At the extreme level, we can set the value of k to the number of instance in the dataset. This division is called *leave-one-out* cross validation. 5-fold cross validation is visualized in Figure A.10.

The advantage of k-fold cross validation is that the entire dataset will be used for testing, so the algorithm will be evaluated more intensively and the dataset is utilized better. On the other side, the time for the experiment will be increased k times.

A.2.3.2 Metrics

Several metrics are defined to evaluate a classification algorithm. In this thesis, we will use primarily *accuracy* and *AUC* [Huang and Ling, 2005].

Accuracy score is simply defined as

$$accuracy = \frac{correct\_prediction\_number}{total\_prediction\_number} \tag{A.18}$$

Accuracy score is used because of several reasons:

- In the case of balanced dataset, the *accuracy* score is the most commonly metric used[Galar et al., 2012]. In this thesis, all Wikipedia datasets used for evaluation are balanced datasets.
- The score is understandable even for normal users [Japkowicz and Shah, 2011]. This feature is not highly prioritized before, but recent studies in interpreting machine learning [M. T. Ribeiro et al., 2016] emphasised the importance of let users understand the algorithm.

However, as studies [Huang and Ling, 2005] have pointed out, *accuracy* does not represent completely the performance of a classification algorithm. Accuracy score tell us the final result of a classifier, but does not tell us how the algorithm behave when the threshold changes.

Therefore, we also used *AUC*, stands for ***A**rea **U**nder **C**urve*. *AUC* is visualized in Figure A.11. In order to calculate *AUC*, we plot *ROC* curve. *ROC* stands for ***R**eceiver **O**perating **C**haracteristic*, with *x-axis* is the false positive rate, and *y-axis* is the true positive rate.

The idea is, we want to observe how the true positive rate of the algorithm changes according to the movement of the false positive rate.

Let's consider an example of binary classification. In fact, a machine learning algorithm in general [Shalev-Shwartz and Ben-David, 2014] does not return a prediction as a hard Positive or Negative value, but will return a numeric value which can be normalized into the range $[0, 1]$ to state the probability that an instance of testing data should belong to a certain class. In a naive approach, one can use 0.5 as the threshold, means that if the return value is less than 0.5 we will classify the instance as Negative, and otherwise we will classify it as Positive. However, based on specific requirements the threshold might be changed.

To calculate *AUC* value, we will vary the threshold from 0 to 1. Then for each threshold value we will plot a point with the coordination as the true positive rate and false positive rate. The *ROC* curve is the set of these points. After plotting the *ROC* curve, we measure the area covered by the curve to receive the value of *AUC*.

The true positive rate (TPR) is defined as:

$$TPR = \frac{TP}{P} \tag{A.19}$$

The false positive rate (FPR) as:

$$FPR = \frac{FP}{N} \tag{A.20}$$

wherein, TP is the number of cases that the algorithm predicted as positive and they are indeed positive, FP is the number of cases when the algorithm predicted as positive but they are in fact negative, P and N are total number of positive and negative case in the testing dataset respectively.

Originally, *AUC* is defined for binary classification problem. There is no standard way to extend the *AUC* calculation to multi-class classification problem. However, in consistent with

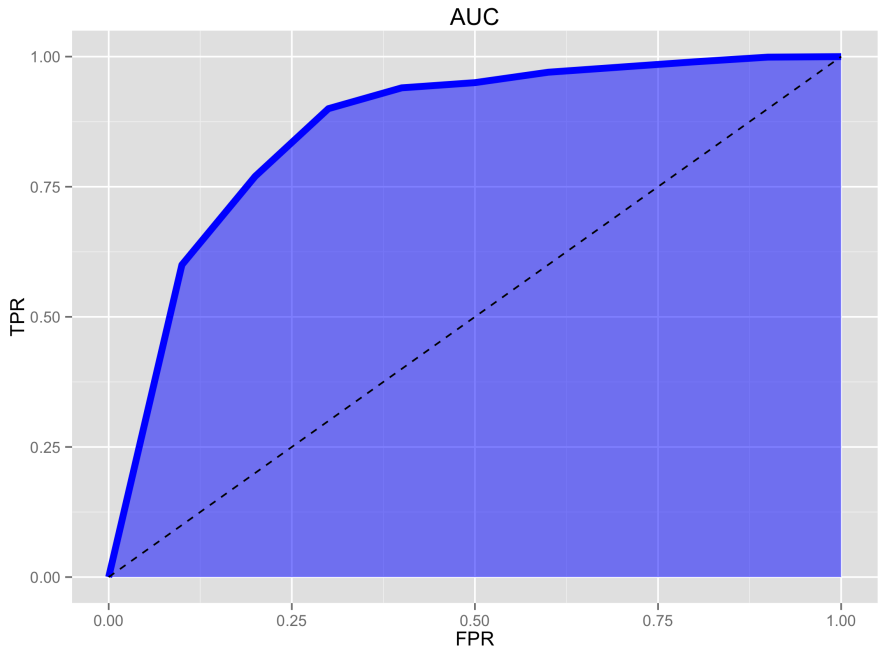


Figure A.11: ROC AUC for binary classification.

previous studies [Q. Dang and C. Ignat, 2016b], we applied the calculation proposed by [Hand and Till, 2001] which is widely used in practice. In compare to other methods, the advantage of the method of [Hand and Till, 2001] is that it produces only a single output value, make it easier to compare between different algorithms.

The AUC by definition of [Hand and Till, 2001] is defined as follow. Given a multi-class classification problem with c class, labelled as 0, 1, ... $c-1$ with $c > 2$. We define $\hat{A}(i|j)$ as the probability that a randomly drawn member of class j will have a lower estimated probability of belonging to class i than a random member of class i. Then we define $\hat{A}(i,j) = \frac{\hat{A}(i|j)+\hat{A}(j|i)}{2}$. The AUC value is calculated as:

$$AUC = \frac{2}{c(c-1)} \sum_{i<j} \hat{A}(i,j) \tag{A.21}$$

Both *accuracy* and AUC values range from 0.0 to 1.0. A higher value means a better algorithm.

Résumé

Les systèmes collaboratifs à large échelle, où un grand nombre d'utilisateurs collaborent pour réaliser une tâche partagée, attirent beaucoup l'attention des milieux industriels et académiques. Bien que la confiance soit un facteur primordial pour le succès d'une telle collaboration, il est difficile pour les utilisateurs finaux d'évaluer manuellement le niveau de confiance envers chaque partenaire. Dans cette thèse, nous étudions le problème de l'évaluation de la confiance et cherchons à concevoir un modèle de confiance informatique dédiés aux systèmes collaboratifs.

Nos travaux s'organisent autour des trois questions de recherche suivantes.

1. **Quel est l'effet du déploiement d'un modèle de confiance et de la représentation aux utilisateurs des scores obtenus pour chaque partenaire ?** Nous avons conçu et organisé une expérience utilisateur basée sur le jeu de confiance qui est un protocole d'échange d'argent en environnement contrôlé dans lequel nous avons introduit des notes de confiance pour les utilisateurs. L'analyse détaillée du comportement des utilisateurs montre que: (i) la présentation d'un score de confiance aux utilisateurs encourage la collaboration entre eux de manière significative, et ce, à un niveau similaire à celui de l'affichage du surnom des participants, et (ii) les utilisateurs se conforment au score de confiance dans leur prise de décision concernant l'échange monétaire. Les résultats suggèrent donc qu'un modèle de confiance peut être déployé dans les systèmes collaboratifs afin d'assister les utilisateurs.

2. **Comment calculer le score de confiance entre des utilisateurs qui ont déjà collaboré ?** Nous avons conçu un modèle de confiance pour les jeux de confiance répétés qui calcule les scores de confiance des utilisateurs en fonction de leur comportement passé. Nous avons validé notre modèle de confiance en relativement à: (i) des données simulées, (ii) de l'opinion humaine et (iii) des données expérimentales réelles. Nous avons appliqué notre modèle de confiance à Wikipédia en utilisant la qualité des articles de Wikipédia comme mesure de contribution. Nous avons proposé trois algorithmes d'apprentissage automatique pour évaluer la qualité des articles de Wikipédia: l'un est basé sur une forêt d'arbres décisionnels tandis que les deux autres sont basés sur des méthodes d'apprentissage profond.

3. **Comment prédire la relation de confiance entre des utilisateurs qui n'ont pas encore interagi ?** Etant donné un réseau dans lequel les liens représentent les relations de confiance/défiance entre utilisateurs, nous cherchons à prévoir les relations futures. Nous avons proposé un algorithme qui prend en compte les informations temporelles relatives à l'établissement des liens dans le réseau pour prédire la relation future de confiance/défiance des utilisateurs. L'algorithme proposé surpasse les approches de la littérature pour des jeux de données réels provenant de réseaux sociaux dirigés et signés.

Mots-clés: collaboration, confiance, théorie des jeux, apprentissage automatique

Abstract

Large-scale collaborative systems wherein a large number of users collaborate to perform a shared task attract a lot of attention from both academic and industry. Trust is an important factor for the success of a large-scale collaboration. It is difficult for end-users to manually assess the trust level of each partner in this collaboration. We study the trust assessment problem and aim to design a computational trust model for collaborative systems.

We focused on three research questions.

1. **What is the effect of deploying a trust model and showing trust scores of partners to users?** We designed and organized a user-experiment based on trust game, a well-known money-exchange lab-control protocol, wherein we introduced user trust scores. Our comprehensive analysis on user behavior proved that: (i) showing trust score to users encourages collaboration between them significantly at a similar level with showing nickname, and (ii) users follow the trust score in decision-making. The results suggest that a trust model can be deployed in collaborative systems to assist users.

2. **How to calculate trust score between users that experienced a collaboration?** We designed a trust model for repeated trust game that computes user trust scores based on their past behavior. We validated our trust model against: (i) simulated data, (ii) human opinion, and (iii) real-world experimental data. We extended our trust model to Wikipedia based on user contributions to the quality of the edited Wikipedia articles. We proposed three machine learning approaches to assess the quality of Wikipedia articles: the first one based on random forest with manually-designed features while the other two ones based on deep learning methods.

3. **How to predict trust relation between users that did not interact in the past?** Given a network in which the links represent the trust/distrust relations between users, we aim to predict future relations. We proposed an algorithm that takes into account the established time information of the links in the network to predict future user trust/distrust relationships. Our algorithm outperforms state-of-the-art approaches on real-world signed directed social network datasets.

Keywords: collaboration, trust, game theory, machine learning