TUTORING THE ELDERLY ON THE USE OF RECOMMENDING SYSTEMS

Anastasios Savvopoulos and Maria Virvou
Department of Informatics
University of Piraeus
Greece

Abstract

Recommending systems are used by many researches to suggest to users all kinds of products. However, not all users of all ages can use these systems. This is particularly the case for the elderly who are not familiar with the computer technology. Moreover, the terminology of recommending systems or user interfaces may prove many times to be a difficult challenge for elderly users. In view of the above, we have created an intelligent tutoring component for product recommending applications. This tutoring component has been created especially for the elderly and it was incorporated into an e-shop application for the interactive TV, called iTVMobi.

Introduction

Nowadays internet, mobile phones, interactive TV and other media have become popular means for various every day activities such as entertainment and shopping. Intelligent systems, created for suggesting entertainment and shopping solutions, have been constructed in order to help users choose appropriate products for them. These systems are called recommending systems. Many times these systems have proved to be quite successful and have helped users choose what they really liked. However, not all users of all ages can use these systems. This is particularly the case for the elderly who are not familiar with the computer technology. Moreover, the terminology of recommending systems or user interfaces may prove many times to be a difficult challenge for elderly users. In view of the above, we have created an intelligent tutoring component for product recommending applications. This tutoring component has been created especially for the elderly. It can be incorporated into any kind of recommending products system and its role is to tutor and help elderly users of recommender systems in an adaptive way. Its reasoning system does not depend on products characteristics so its product independent. It is also medium independent as it is a separate component that functions outside the main reasoning of the system that incorporates it. This component helps elderly users use product recommending systems and also predicts mistakes made by the elderly based on this group’s most popular disadvantages such impaired sight, hearing and lack of understanding.

There are many examples of intelligent tutoring systems in e-learning, like the work of Frasson et al., 1997. Their work uses pedagogical agents to in a multi strategic tutoring system. Their paper describes the use of actors for implementing pedagogical strategies and more generally for detecting which strategy is more suited to a given learner. Their approach leads to the definition of a multi-strategic ITS based on pedagogical actors, that is able to switch among various strategies. In this framework pedagogical actors are used to model the expertise of the various pedagogical strategies. Another important work on the same field has been done by Heffernan and Koedinger (2002). Their work presents Ms. Lindquist, an Intelligent Tutoring System (ITS) designed to carry on a tutorial dialog about symbolization. Ms. Lindquist has a separate tutorial model encoding pedagogical content knowledge in the form of different tutorial strategies, which were partially developed by observing an experienced human tutor. A very important work has also been done by Suraweera and Mitrovic (2004). Their paper presents KERMIT, a Knowledge-based Entity Relationship Modelling Intelligent Tutor. KERMIT is a problem-solving environment for the university-level students, in which they can practice conceptual database design using the Entity-Relationship data model. KERMIT uses Constraint-Based Modelling (CBM) to model the domain knowledge and generate student models. We have used CBM previously in tutors that teach SQL and English punctuation rules.

Significant work has also been done by Brusilovsky et al. (1996) with ELM-ART. Their research discusses the problems of developing WWW-available ITS and, in particular, the problem of porting existing ITS to a WWW platform. They present the system ELMART which is a WWW-based ITS to support learning programming in Lisp. ELM-ART demonstrates how several known ITS technologies can be implemented in WWW context. Another important work in web ITS has also been done by Tsiriga and Virvou (2004). In their work they describe a framework for the initialization of student models in Web-based educational applications. The framework is called ISM. The basic idea of ISMis to set initial values for all aspects of student models using an innovative combination of stereotypes and the distance weighted k-nearest neighbour algorithm. In particular, a student is first assigned to a stereotype category concerning her/his knowledge level of the domain being taught. Then, the model of the new student is initialized by applying the distance weighted k-nearest neighbour algorithm among the students that belong to the same stereotype category with the new student.

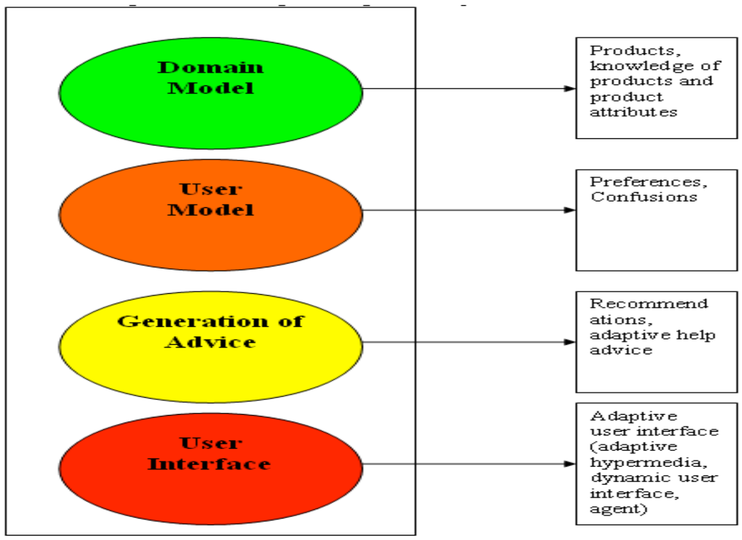
Last but not least we have two different approaches on the field of e-learning ITS. The first research has been done by Graesser et al. (2005). Their work presents AutoTutor that simulates a human tutor by holding a conversation with the learner in natural language. The dialogue is augmented by an animated conversational agent and three-dimensional (3-D) interactive simulations in order to enhance the learner’s engagement and the depth of the learning. And the second work has been done by Baker et al. (2006) and introduces a system which gives a gaming student supplementary exercises focused on exactly the material the student bypassed by gaming, and which also expresses negative emotion to gaming students through an animated agent. Students using this system engage in less gaming, and students who receive many supplemental exercises have considerably better learning than is associated with gaming in the control condition or prior studies.

The novelty of our approach compared to the researches above lies in three reasons. The first is the entirely different field that we apply the ITS general architecture, which the field of e-shopping. The second reason is that the user group targeted by our application is a special group of users that due to their age they have impairment concerning the interaction with computer applications and knowledge of the product itself. The third reason is that the mechanism of generating advice in our system is a combination of technologies from the above system. More specifically, our system uses a 3-D animated agent, adaptive hypermedia and a dynamic user interface.

The Intelligent Tutoring Component Architecture

The main architecture of a tutoring system involves four major parts (Figure 1 left). The first is the Domain Model. This model includes all the information about the domain of the e-learning system, in other words what the intelligent tutoring system tries to teach to the student. The Domain Model includes the knowledge of the teaching field and also all the characteristics concerning this knowledge. Such characteristics can be area difficulty, knowledge level etc. In our approach the Domain Knowledge involves all the information about the product and their specific attributes. More specifically, the Domain Model includes information about how much the customers off all ages are familiar with the product, technical characteristics and also special attributes of the products that target specific age groups. Moreover, because our tutoring system targets the group of elderly people specific information about products attributes that concern this age group are stored in the Domain Model. The second part of the architecture of a tutoring system is the User Model. In this part all the information about students or users in general are stored in order for a user model to be created for every specific user. This information may include stereotypical data from user stereotypes, explicit information from answering questionnaires or implicit information from observing users' behavior. In our case we have two kinds of information. This first involves prefaces and interest degrees on the products that the application tries to sell. The second kind involves mistakes and confusion degrees concerning the usage of the system.

Figure 1: From the General Tutoring System Architecture to Our Approach



The third part of a tutoring system is the Generation of Advice. In this part the tutoring system based on information from the user model and the domain model makes assumptions and tries to resolve problematic situations. For example, if a student does many mistakes and concludes his/her tests in very little time then it can assume the particular student is careless. However, this part does not stop there but continues and tries to resolve the problem, maybe by suggesting to the student to take more time to conclude his/her test. In our tutoring system we have two kinds of suggestions extracted from the information observed by the customer's behavior. Our system observes customer's behavior and follows the buggy approach (Tsiriga & Virvou, 2004). The buggy approach means that the system assumes what the customer should buy or should do while interacting with the system. The Generation of Advice in our tutoring system is consisted of two kinds of advices. The first kind is recommendation of products that the system assumes a customer should buy based on his/her moves and the second kind is adaptive help actions that try to help the customer on what should do concerning a mistake with his/her interaction with the application. The last part of a tutoring system concerns the presentation of advices. There many technologies in intelligent systems that someone can customize the presentation of advices, but all of involve the user interface. In our approach we followed a combination of such technologies to achieve an adaptive user interface. The first technology is adaptive hypermedia, the second dynamic user interface adaptivity and the third is an animated agent. These three technologies are better explained in our case study. Combining these three technologies we managed to create an adaptive user interface that changes according customer preferences and confusions. Our tutoring system follows this general architecture of tutoring system but transfers the tutoring systems' techniques to a novel area, the area of e-commerce. Moreover, the age group that our system targets, which is the elderly people, creates an even more difficult environment for a system to be effective on tutoring.

The Case Study

In order to test our approach we incorporated it in an e-shop application that sells phones called iTVMobi. iTVMobi is an adaptive mobile shop created for the interactive television that learns from customer preferences (Figure 1, Figure 2). ITVMobi was built on Microsoft TV (MSTV) technology. The core components of MSTV are available in the Windows XP operating system and can be run on personal computers (PCs). MSTV technology can be utilized within a familiar and mature Integrated Development Environment (IDE). Microsoft Visual Studio offers a multitude of tools for designing, developing, testing and deploying an application. ITVMobi can be used by a telemarketing channel to sell mobile phones in a personalized way. Its aim is to provide help to customers with hearing and sight problems by suggesting the best mobile phone for them. The recommender system that makes suggestions concerning mobile phones and accessories is based on user modeling. The system can learn about the users' preferences and provide more helpful responses. User models are created using clustering algorithms. These techniques will be explained more thoroughly in the next section.

For every user, iTVMobi creates a different record at the user model database. In iTVMobi every customer can visit several mobile phones. For the purposes of our research we have implemented the system for five popular mobile brands. Every customer has her/his own personal shopping cart. If customers intend to buy a phone they must simply move the phone into their cart by pressing the specific button or they can press the buy button at their remote control at the time that the specific product is shown on their TV screen. They also have the ability to remove one or more phones from their cart by choosing to delete them. After deciding which phones to buy, a customer can easily purchase them by pressing the button "buy" at their shopping cart. All navigational moves of a customer are made through the TV remote control and are recorded by the system in the statistics database. In this way iTVMobi saves statistics considering the visits in the different brands and specific phones individually. The same type of statistics is saved for every customer and every phone that is moved to the buyer's cart. The same task is conducted for the mobile phones that are eventually bought by every customer. All of these statistical results are scaled to the unit interval [0, 1].

In particular, ITVMobi interprets users' actions in a way that results into two different functions. The first is the calculation of users' interests in individual phones and production companies and the second is the interpretation of users' actions concerning possible navigation mistakes. Each user's action contributes to the individual user profile by showing degrees of interest into one or another company or individual phone or by showing likelihood on a specific mistake. For example, the visit of a user into a phone-icon shows interest of this user to the particular phone and its brand. If the user puts this phone into the shopping cart this shows more interest in the particular phone and its brand. If a user buys this phone then this shows even more interest whereas if the user takes it out of the shopping cart before payment then there is not any increase in the interest counter. On the other hand if a user follows a different pattern of navigational moves, like repeated clicks on the same brand-name, the system interprets this action but as a confusion navigational mistake rather than as a high degree of interest in this brand. Thus, in this case, the system decides to intervene with an adaptive help action.

Apart from brands that are already presented, other features that are taken into consideration by iTVMobi, for customer interest degree, are the following: phone price range, phone technical features, phone size, phone connectivity, phone display features, phone memory capabilities and phone battery autonomy. All the above phone features are measured in three degrees. For example size can be small, medium or large. Every different feature is consisted of several feature values. Size is consisted of dimensions and grams. The technical features are consisted of phone functions, operating system, java abilities etc. The phone connectivity is consisted of bluetooth, infrared, gps and wlan abilities. The phone display is consisted of screen size, pixels and colors that can showed in the phone screen. The phone memory is consisted of internal memory, expandability or hard disk of phone. Lastly, phone battery is consisted of talk time and stand-by time. The price of every phone belongs to one of the five price ranges: 100 to 250 €, 251 to 400 €, 401 to 600 €, 601 to 800 € and over 801€. As for mistakes degrees we consider the following: difficulty of the user with sight and hearing problems to see brands' names, difficulty to see phones' names and pictures and the confusion degree. Suggested phones are presented in the suggestions window through the help of adaptive hypermedia. Moreover, iTVMobi uses an animated agent to inform and help the users throughout the system. This agent can give information about the usage of the system if a user cannot understand some sections.

The User Model

In order to achieve adaptivity in iTVMobi we used a user modelling technique based on features and problems of our customers. In this section we will present the user modelling process in general. We will emphasize on the general steps of this process and not on how this process is implemented in the two different systems. Further details on these two implementations will be given in the later sections.

The user modelling process is consisted of five major steps. At first, we conduct the Data Acquisition Step. In this step we define the data acquired, how this data will be acquired and in what form this data will be saved in order to be efficient for the next step of the user modelling process. For the data definitions we agreed that all data will be rates of either customer needs, interests, problems or mistakes. The data concerning needs and interests are based on the product features and concern visits on product pages, product acquisitions, actions with the shopping cart and request of product videos. The data concerning problems and mistakes are based on navigation problems, hearing and seeing disabilities and concern real time navigation problems and mistakes in product pages and shopping cart.

This data can be acquired by two different ways, explicitly or implicitly. Explicitly a customer can answer questions concerning tastes or problems, rate products, product categories or even other customers. Implicitly the system observers navigational behaviour (visits in product pages, buying products, making wrong navigation patters) and makes assumptions that alter this customer's user model. This data is saved in rate form in a database of user models in vector format in order to be easily available for the next of the user modelling process.

The next step of this process is Group Formulation. In this step the system uses a clustering algorithm in order to form groups of similar users (users with similar tastes for the recommender and user with similar mistakes for adaptive help). The input date is takes by the previous step and is inserted in the clustering algorithm. The clustering algorithm we chose to implement is k-means. The clustering algorithm is used to provide clusters groups of similar users that have a representative vector that does not necessarily is one the old users. This process is conducted in the same way for both functions, the recommender and adaptive help, but separately for every function. At the end of this step we have two results for every of the two functions. The first result is the groups of similar users and the second result are the representatives of these groups. The groups are used by the system for the dynamic stereotypes and the community system and representatives are used for the system's adaptive actions and responses (recommendations and help actions).

Next step is User Model Creation. In this step iTVMobi uses groups and representatives to create dynamic stereotypes. For the users that iTVMobi has few information creates a user model that combines stereotypical information and individual information. In this way iTVMobi fills rates about a user's interests or mistakes from the stereotypical in order to make assumptions about this specific user. On the other hand, if iTVMobi has enough information about a user then creates an individual user model based on previous user explicit and implicit data, group and representative. In this way every time the user model of this specific user is created dynamically but based on previous data and user model of this user. The next step is Adaptivity. This step incorporates all actions that iTVMobi takes in order to recommend products, change its interface dynamically or help users adaptively. In this step iTVMobi can use many techniques in order to provide product recommendations, help every user adaptively or correct users' mistakes. The techniques that iTVMobi uses are adaptive hypermedia, dynamic user interface alteration and an adaptive animated help and recommending agent. These techniques help iTVMobi to change the user interface adaptively in order to recommend products or catch user mistakes and provide the user with solutions or help the user navigate easily. In the next two sections we explain the major two functions that iTVMobi uses to help users adaptively choose easily a product that suits their interests.

Generation of Advice: Product Recommendations

The recommender system is based on user modeling that is constructed using the k-means algorithm. The recommender function is based on the principle that many customers tend to have similar interests. Every customer's interest in one of the phone features described above is recorded as a percentage of his/her visits in the respective phone pages. An interest of the customer at a particular phone is calculated by the equations 2 to 5.

$$InterestInFeature_1 = \frac{VisitsInPhonesWithThisFeature}{VisitsInAllPhones} \quad (2)$$

$$InterestInFeature_2 = \frac{PhonesPlacedInBasketWithThisFeature}{AllPhonesPlacedBasket} \quad (3)$$

$$InterestInFeature_3 = \frac{PhonesBoughtWithThisFeature}{AllBoughtPhones} \quad (4)$$

$$\begin{aligned} InterestInFeature = {} & W_{C1} * InterestInFeature_1 + \\ & W_{C2} * InterestInFeature_2 + W_{C3} * InterestInFeature_3 \end{aligned} \quad (5)$$

As the previous equations show the degree of interest in a phone feature is measured in three ways. Then in order for the full degree of interest to be acquired the system calculates a weighted sum of the three different degrees of interest, the degree of interest that corresponds to the visits of the user in the phone pages, the degree of interest that corresponds to the phones placed by the user to his/her basket and the interest that corresponds to the phones bought by the user. The weights used by the system are different for every phone feature and were extracted through the experience from the evaluation process of the system. For example a user chooses to visit a phone through the phone icon and does not have the ability to know the display abilities of this phone before opening the specific phone page. As such, the opening of a specific phone page through its icon may not mean that the user is necessarily interested in the phone but that s/he is just browsing several phones. On the other hand every company's name is displayed from the very beginning to every user and in this way the user is aware for the company that he/she selects to visit thus making his/her selection more accountable. As a result the W_{C1} weight used to measure the Interest in Company from the user visits is bigger than the weight W_{D1} used to measure the Interest in Phone Display from user visits in different phones. The recommender module uses the k-means clustering algorithm in order to create representatives of customer groups that the system uses to make buying suggestions. The recommender takes as input the statistical data, described above, of the navigational moves of every customer and feeds them to the clustering algorithm. The clustering algorithm provides the recommender with clusters-groups of customer that have similar tastes. The recommender module takes these results and calculates the representatives of every group.

Every time a customer uses the system the recommender module finds his/her representative and proposes phones based on the representative taste percentages through the use of adaptive hypermedia. After creating the proposing phones list, the recommender system considers the mistakes statistics database and chooses a corresponding list of accessories. These accessories are combined with proposed phones list in order to provide the customers with hearing and sight problems with a more complete solution for their needs. For example if the recommender finds a phone that is very close to the representative's tastes than this phone is noted as "recommended" product and is given a different type of indicator than a phone that is more far, considering the tastes of the representative. Then the system finds an accessory corresponding to this phone and to the mistakes made from this user. If a new user enters the system the recommender classifies him/her to the group that has the largest number of members. This is based on the idea that if many users have similar tastes then a new user is probably going to have similar tastes and mistakes with the majority of them. The degree of recommendation is presented through adaptive hypermedia. The product that has the highest degree of interest for this user is noted as a "recommended" product and the one with a lower degree is noted as "check this too" product. Similar degrees of annotation are used for the corresponding accessories. Sample screenshots of the recommendation page are illustrated in the figures below.

Figure 4 left: Sample screenshot of the two annotations, "hot product" (left) and "check this" (right). **Figure 4 right**: Screenshot from the phone and accessories recommendations.

Generation of Advice: Adaptive Help Actions

The adaptive help module concerns the adaptive help responses. This module tries to identify mistakes in the navigational moves of every user. This module is based on the principle that many users with sight and hearing problems tend to have similar navigational mistakes. Again, the k-means algorithm is used to group users but in this case a different set of input data is used. The input data consists of the mistake degrees that were introduced in the above section. Mistakes are considered as different "wrong" navigational patterns. For example, a user can make "confusion navigation" like the continuous visiting of two neighboring production company buttons. This action raises the possibility of vision problem. Another example is "navigation without a purpose" witch can be achieved by a pattern of pressed buttons and clicked areas that leads to no purpose. This action raises the confusion problem. Degrees are calculated as a percentage of specific mistakes committed in a specific phones' page. For example the disability to see companies' buttons degree is calculated by equation 9 and disability to recognize phone icons is calculated by equation 10.

$$HardtoSeeCompany = \frac{MistakesInCompaniesButtons}{TimesInCompaniesPage} \quad (9)$$

$$HardtoSeePhoneIcons = \frac{MistakesInPhoneIcons}{TimesInPhonePages} \quad (10)$$

If a user has many navigational mistakes then the system responds and tries to help this user with help actions customized to his/her mistakes. These actions can vary a lot. For example, the "confusion navigation" results in actions such as the automatic changing of the size of brand names buttons or phone links. This can result in a clearer presentation of the user interface. Another action taken by the system is changing the location of brand names' buttons in order to avoid confusion. Other actions involve speech synthesizers and agents pointing on the screen in order to help customers understand the locations of the user's interface components. The "navigation without a purpose" can result in actions like showing an options message and asking the user directly what he/she wants to do. An example of a wrong navigational pattern can be the following: a user chooses to click on company button, then click the adjacent company button, then click the previous company again and then click the same adjacent company again. These four moves are interpreted by the system as a possible mistake of confusion between company buttons. Every time a customer uses the system the adaptive help system finds his/her representative and responds with adaptive help actions.

User Interface: The Adaptive User Interface

The adaptive user interface of iTVMobi can change the user interface in real time according to the specific customer mistakes. An example can be seen in Figures 5 and 6. In this particular example the system observes the user's navigation moves between two neighboring mobile phones and counts his/her mistakes. If a user has made a lot mistakes in this section, like browsing two neighboring phones repeatedly without putting any of phones in his cart at the meantime, then the system identifies that the user cannot view the phone pictures clearly and chooses to enlarge them. If the mistakes between the two neighboring phones continue then the system identifies that the user has confused only these two phones. The action taken by the system is to change the location of these two phones and move the one away from the other, while bringing a different phone close in order not to destroy the whole arrangement in the screen of the phones. If the user continues to make the same kind of mistakes then the system uses the animated agent in order point the phones by moving next to them, showing them with its "hand" and then telling with its "voice" the model of the phone. The system also increases the sound volume in order to help people with hearing problems understand more clearly the point out function of the animated agent. If the user finds annoying the changes of the user interface than he can disable them from his profile page.

Figure 5 Left: First stage of the phone user interface. Showing small pictures of mobile phones. **Figure 5 Right:** Second stage of the phone user interface. The user has made mistakes. Bigger phone pictures and a next button showing that phones are split in two pages.

Figure 6 left: Third stage of the phone user interface. The user has confused the first two phones on the bottom. The system has changed their locations and brought the silver phone near the first phone on the bottom. **Figure 6 right:** Fourth stage of the phone user interface. The user continues to confuse the phones. The system enables the animated agent in order to point the phones and increases the sound volume the agent.

Conclusions

In this paper we presented an intelligent tutoring component for product recommending applications. This tutoring component has been created especially for the elderly. It can be incorporated into any kind of recommending products system and its role is to tutor and help elderly users of recommender systems in an adaptive way. Its reasoning system does not depend on products characteristics so its product independent. It is also medium independent as it is a separate component that functions outside the main reasoning of the system that incorporates it. This component helps elderly users use product recommending systems and also predicts mistakes made by the elderly based on this group's most popular disadvantages such impaired sight, hearing and lack of understanding. We incorporated the component mentioned in a case study, an interactive TV shop that incorporates the intelligent tutoring component.

References

Baker, R., Corbett, A., Koedinger, K. R., Evenson, S., Roll, I., Wagner, A., et al. (2006). Adapting to when students game an intelligent tutoring system. In M. Ikeda, K. D. Ashley, & T.-W. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (pp. 392–401). Berlin: Springer.

Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In C. Frasson, G. Gauthier, & A. Lesgold, A. (Eds.), *Proceedings of the Third International Conference on Intelligent Tutoring Systems, ITS-96* (pp. 261–269). Berlin: Springer.

Frasson, C., Mengelle, T., & Aimeur, E. (1997). Using pedagogical agents in a multi-strategic intelligent tutoring system. In *Proceedings of the AI-ED '97 Workshop on Pedagogical Agents*, (pp. 40–47).

Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005a). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions in Education, 48,* 612–618.

Heffernan, N. T., & Koedinger, K. R. (2002). An intelligent tutoring system incorporating a model of an experienced tutor. In S. A. Cerri, G. Gouardères, & F. Paraguaçu (Eds.), *Intelligent Tutoring Systems* (pp. 596–608). Berlin: Springer.

Suraweera, P., & Mitrovic, A. An intelligent tutoring system for entity relationship modelling. *International Journal of Artificial Intelligent in Education, 14* (3-4), 375–417.

Tsiriga, V., & Virvou, M. (2004). A framework for the initialization of student models in web-based intelligent tutoring systems. *User Modeling and User-Adapted Interaction, 14*(4), 289–316.