INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

A Web-Based Experiment on Dialogue Summarisation

N. T. Roman *P. Piwek*
A. M. B. R. Carvalho

Technical Report - IC-05-05 - Relatório Técnico

March - 2005 - Março

A Web-Based Experiment on Dialogue Summarisation[*]

Norton Trevisan Roman[†] Paul Piwek[‡]
Ariadne Maria Brito Rizzoni Carvalho[§]

Abstract

This document describes technical details of a web experiment carried out at the State University of Campinas, Brazil, on summarisation of a set of dialogues. Our intention here is to present the strategy we followed to build the experiment's website, as well as the statistical issues and the technicalities involved. In this report, we do not present the experiment's results.

1 Introduction

In this document, we describe technical details of an experiment carried out on the web at the State University of Campinas, Brazil, from September 17 to November 01, 2004. Our main goal is to describe the experiment's strategy, along with the algorithms used in the webpages.

In this experiment, participants had to summarise four movie dialogue scripts, one at a time. This process should be done in one single go, and the participants were not allowed to go back to previous pages to change their summaries.

The dialogues took place between vendors (or servants) and customers (or clients). In two of them, either one or both dialogue partners were impolite, while in the remaining two, they were polite. This characteristic increased the number of experimental conditions we had to deal with. See Section 2.3 for more details.

Participants where instructed to summarise the dialogues according to a given point of view: either customer, vendor or observer. To some of the participants a maximum summary length, corresponding to 10% of the number of words in the dialogue, was imposed, while no limit was imposed to the rest, *i.e.*, it was up to the participants to decide how many words to use in the summary.

The rest of the paper is organised as follows. Section 2 describes the methodology adopted in this experiment; Section 3 shows the implementation details; Section 4 presents

[*]We would like to thank CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico – and CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – for sponsoring the research. Part of it was also supported by the EC Project NECA IST-2000-28580.

[†]Institute of Computing, Unicamp, Brazil

[‡]Information Technology Research Institute, University of Brighton, UK

[§]Institute of Computing, Unicamp, Brazil the results we obtained from adopting such a methodology; and, finally, Section 5 presents our conclusions.

2 Methodology

To reduce drop-out and its negative impact, the experiment followed the Hard-Hurdle and Warm-Up techniques, as presented in [3, 1, 4, 5], and the Wextor Project [6], as well as other techniques, also presented in [7]. Figure 1 illustrates the adopted design.

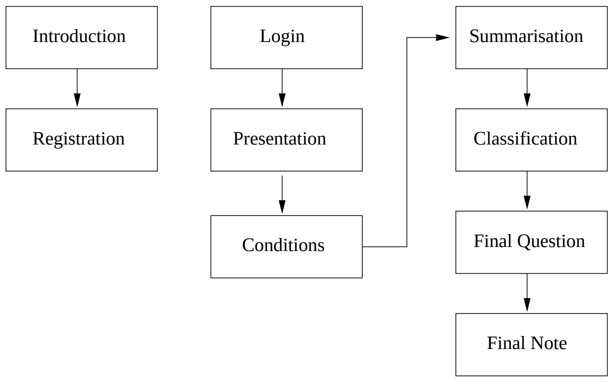

Figure 1: Experimental Design

In the first four steps, from the Introduction to the Presentation phase, the experiment's design mirrored the experimental design described in [7]. Briefly, in the Introduction stage, participants went through an introductory text explaining the set-up of the experiment, without giving away the real goal. The duration of the experiment was also informed in this text. In the Registration step, participants were asked for some personal information and were given, through e-mail, a password, so that they could log-in the next stage (Login). From the Login stage participants were sent to the Presentation stage, where they found yet another text, giving some historical background to the research.

The next four stages define the real experiment, *i.e.*, they do not function primarily as a measure to reduce drop-out. In these stages, participants were initially randomly distributed amongst the experimental conditions (Conditions stage). In the Summarisation stage, participants started summarising the dialogues. As an extra precaution, after participants summarised all the dialogues, they were asked to classify the same dialogues according to their politeness degree (Classification stage). This allowed us to compare the participants' own perception of the dialogues with data we gathered in a previous experiment on the politeness degree of the dialogues [7].

As all the dialogues were taken from movie scripts, there was a possibility that participants would recognise them and, if so, there would be a threat to the experiment's validity, for participants might bring to the summary information not present in the dialogue (coming from the way they recall the whole scene). In order to detect such confounding, we introduced the Final Question Stage, where we asked the participants whether they had recognised any of the dialogues and, if so, from where.

As an incentive, we announced that any participant who successfully completed the experiment, following the instructions, would be eligible for one out of two prizes of R$100,00. This measure was taken as an attempt to reduce drop-out [1].

Finally, in the last stage (Final Note), we presented the participants with a thank you note. We also presented a web address where they could check the prize winners, and an e-mail address, in case they needed some extra information or wanted to comment on something.

The participants were instructed to complete the experiment in one go. To guarantee this, we took some steps to prevent a participant already logged in to leave and log in again. Other steps that we had to take were to prevent the pages, apart from the initial one and the login page, to be directly accessed. These pages had to be accessed in a very specific order (see Section 3 for details).

Participants were also not allowed to change their previous answer. Once they had summarised a dialogue, they could not change it and, had they tried to do so, the system would have blocked them from doing so. We stressed this fact in all the instructional texts.

2.1 Materials: the Dialogues

In this experiment we used four dialogues, extracted from movies scripts, portraying interactions between a customer and a vendor or servant. These dialogues were translated to Portuguese, to better suit our needs. Whenever necessary, they were also modified to fit the required characteristics of being a two-persons dialogue and having no additional information apart from the utterances, like in "Client: <angry> I won't!" or "Angry Client: I won't!", which are very common in movie scripts.

Two of the dialogues portray an interaction where both parties are polite, while in the two remaining dialogues some of the parties are perceived as impolite. The politeness degree of the dialogues was determined by a previous experiment, described in [7].

2.2 Participants

Our participants were graduate and postgraduate students from the State University of Campinas, Brazil. We chose to consider only these students because (1) it was easy to control for false identity issues, (2) there was no cross-cultural confounding, for almost all of them were Brazilians [3], (3) they were quite well distributed according to gender, (4) we could find people from almost all Brazilian states in the university, and (5) the big set of possible participants available – only in 2001, according to the university website[1] there were 11,187 graduate, 4,466 MPhil and 4512 DPhil students, being 20,165 students in total.

One problem about this population is that, if we want to generalise our claims to the whole Brazilian population, the fact that they are university students might result in a non-representative set of the Brazilian population. Therefore, all of our claims are restricted to

[1] $http : //www.aeplan.unicamp.br/anuario\_estatistico\_2002/html/pag037.html$ and $http : //www.prpg.unicamp.br/tabgraf\_alunos.phtml$

the university's subset, i.e., to the population with a high education, mostly from the state of São Paulo.

In order to reach the participants, we broadcasted an e-mail to all the university students, asking them to take part in the experiment. We only mentioned that we were doing an experiment on dialogues, that it was a voluntary work, that it would not take too long, and that those who finished the experiment would be eligible to win a cash prize.

We mentioned nothing else, not even how much the prize would be. Instead, we gave them a link to the experiment's first page (index.html – see Section 3.1) and told them that they could find more information there.

2.3 Variables

In this experiment, we had two between-subjects variables: point of view and summary length. Point of view had three levels, namely, customer, vendor and observer. Summary length, on the other hand, had only two levels: no limit and 10% of the entire dialogue. These variables give rise to six conditions.

The experiment also had one within-subjects variable, *i.e.*, the variable affecting all the participants, which was the order of the dialogues. In order to avoid confounding among the summarisation of the dialogues and the order they were actually presented to the participants, *i.e.*, to avoid order effects [5], when a participant started the experiment, a dialogue order was randomly generated.

The dialogue order was generated according to the Latin Square distribution. As we had two polite and two impolite dialogues, the generated orders where polite-impolite-impolite-polite, impolite-impolite-polite-polite, impolite-polite-polite-impolite and polite-polite-impolite-impolite. Participants were distributed amongst these other four groups. This measure included a between-subjects factor in the experiment, coming from the within-subjects one [4].

The number of groups the participants were distributed into was the combination of all the groups from both between-subjects and within-subjects factors, *i.e.*, 24 groups. The dependent variable, *i.e.*, the variable measured, was the summary of each dialogue, which will be annotated for its politeness content.

2.4 Addressed Issues

In the experimental design, we addressed many important threats to the internal validity of the experiment, like the problems of selection, history and maturation, multiple submissions, instrumentation, experimenter bias, ecological validity, drop-out, technical variance, generalisation, form bias and, finally, misunderstandings. All these issues were described in depth in [7]; please refer to that report for more details.

3 Technical Implementation

The experiment was programmed in PHP3, and required seven webpages, five of them dynamic pages, in order to follow the rules of thumb provided in [3, 1, 4, 5] and the steps in Figure 1. Figure 2 shows the experiment's scheme.

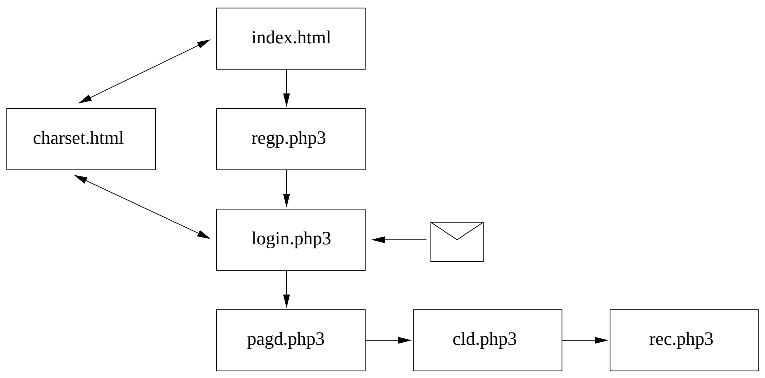

Figure 2: Experiment's Webpages

As soon as participants entered the registration page, a session ID was created. This session ID (hereafter called SID) was sent throughout all the experiment's pages, allowing the script to identify the participant that was submitting the information.

All the experimental data was kept in a hidden folder (under the Unix operating system) – the experiment's folder – called ".Epto". The personal and log information from the participants, along with their submissions in the pages, were kept in a number of subdirectories inside this main folder, one for each participant. Each participant's folder was named after the SID assigned to him/her.

The password file, ".sEpto", was kept inside the experiment's folder. In this file we kept the participant's name, his/her university ID, the encrypted password and SID. The files created inside each participant's folder were:

- drop: created by regp.php3 and updated by every script since. It kept the drop-out information.

- log: created by regp.php3 and updated by login.php3. Kept information about the participant's computer, such as the IP address and port used in the connection, the browser and operating system types, as well as date and time of registration and login.

- .lg: created by login.php3. It signalled that the participant had already logged in, preventing him/her to log in again.

- .infp: created by regp.php3 and updated by pagd.php3. Stored the personal information the participant typed in the registration form, as well as the experimental condition that was randomly assigned to each participant.

- .prim: created by pagd.php3. It was used to signal that the participant had seen the first dialogue in the sequence, preventing him/her from going back through the pages and trying to resubmit, to obtain a different dialogue sequence.

- .r41z, .r42z, .r43z, .r44z: created by pagd.php3. Each of these files kept the summary for the corresponding dialogue, following the sequence they were shown to the participant. Within this design, .r41z corresponds to the first dialogue in the sequence, .r42z to the second, and so on.

- .Cl: created by cld.php3. It kept the classification of the dialogues by the participants.

- .recon: created by rec.php3. It kept the information about whether the participant recognised any of the dialogues s/he summarised.

The experiment's folder contained also a folder named "temp". This folder had one file only, called "cond". It was used by the experimental condition generator to keep track of the conditions it had already given to the participants.

It is worth noticing that the above filenames are not straightforward, neither are those for the experiment's webpages and directories, apart from the pages that should be directly accessed by the participants (index.html and login.php3). This was to protect the experiment's structure from outside observers [4, 5].

Finally, all data from the forms were sent through the POST method. The reason why this method was chosen was to not display any information about the experiment or the participants, as it would be the case if we had chosen the GET method instead [4, 5].

3.1 Index.html

This was the first page of the experiment and the page whose address was sent in the call for participation e-mail. It implemented the "Introduction" phase in Figure 1, showing a presentation text to the participants.

In this text we briefly explained what the participants had to do in the experiment, but without mentioning our real goal. We also mentioned the prize draw and how participants would be eligible for it. The text was a bit long, saying that participants should not go back to previous pages, for they might be dropped out, and that the experiment should take between 20 and 50 minutes [3], depending on the network connection speed. The rationale behind a long text is to have participants with little interest in the experiment drop out [1]. This page was designed following the Hard-Hurdle technique.

We also included in this page a link to "charset.html", which the participant should visit in the case s/he was not able to see graphical accents on the text. At the end of the text, the participant could see an e-mail address to which s/he could send us any enquires.

As a secondary function, this page was necessary to protect the experiment's structure from the outside [4, 5]. If this page did not exist, the participant would be able to see all the files that compose the experiment just by typing the web address.

3.2 Charset.html

This page contained information on how the participants could change their browser's character set in order to see the graphical accents on letters. The information was given for four different browsers: Mozilla, Internet Explorer, Netscape and Epiphany.

3.3 Regp.php3

Figure 3 outlines the algorithm for this script page. Its primary goal was to get some information about the participants, implementing the "Registration" phase in Figure 1. It was designed according to the Hard-Hurdle technique, being responsible for giving to the participant an impression of control by the experimenters and trying to make the participant to get committed to the experiment. In this page we also have implemented a password-dependent access, and created a session ID, in order to avoid and control multiple submissions [5]. See [7] for the session ID generation algorithm.

This page could only be accessed from "index.html". Here the participant was asked to provide some personal information, explaining why we were asking and how we were going to use it. We asked for the participant's name, university ID and a valid e-mail address. We explained that the name and ID were necessary to check the participants identity when receiving the prize, while the e-mail was necessary to send him/her a password to proceed with the experiment. We also said that after the prize distribution was over, these three data would be erased from our records, guaranteeing the data's confidentiality [3].

We also asked for the participant's sex, knowledge area, level (undergraduate etc), age (less than 20, 20 – 25 etc) and the Brazilian state s/he came from. To separate unanswered items from the ones the participant did not intend to answer we included the negative option "I don't want to answer..." and a preselected neutral option ("Choose...") in all the multiple choice items [4]. We said that this information would be used for statistical purposes only.

The system checked the format and content of the submitted information. Had the participant left any data unanswered, the script would tell him/her that s/he forgot to fill in some data [5]. Furthermore, if there were any problem, the script would show an error message highlighting the problematic items in the submitted data on the registration form. All fields were tested for unanswered items and invalid patterns such as an e-mail address without an '@' and a university id not having 6 digits only.

As a last piece of information, we have recorded log files containing the connection information, like the browser and operating system the participant used, the IP address and connection port [5] and the date and time of registration.

3.4 Login.php3

This page followed the very same algorithm described in [7]. Please refer to that work for more information.

3.5 Pagd.php3

This was the page where the experiment actually began, implementing the "Conditions" and "Summarisation" phases in Figure 1. Up to this page, we have tried to drop-out the less motivated participants, hoping that only those who really intended to proceed with the experiment would take part on it.

The algorithm for this script page is shown in Figure 4. Its primary goal was to randomly distribute the participants into the experimental conditions and to show them the dialogues

```
1. Get the Process ID (PID) of the current process.
2. Get the visited flag (VIS) from POST.
3. If this is not the data submission from this page (from regp.php3):
   (a) If this page was not called from "index.html":
         i. Show an error message
   (b) Else:
         i. Generate a Session ID (SID).
        ii. Create a folder for the participant's data, name it with the SID.
       iii. If the folder already exists, give an error message and leave the script.
        iv. Save VIS in the drop-out file, inside the SID folder.
         v. Show the registration form (with the value for SID and the updated VIS value
            being sent as hidden fields).
4. Else:
   (a) Get the personal information from the form, through POST.
   (b) Get SID from POST.
   (c) If there is no problem with the submitted data:
         i. If the participant has already been registered, show an error message and stop
            the script.
        ii. Randomly generate a password, using numbers, capital and lowercase letters,
            and encrypt it.
       iii. If the log file already exists, i.e., the participant went back in the pages and
            tried to resubmit his/her personal data, show an error message and stop the
            script.
        iv. Get the current time.
         v. Create a log file, with connection information, like the client IP address, port,
            browser etc, as well as the current time.
        vi. Lock the password file (see [7]).
       vii. Save the data in the password file. The format of each line in the file is:
            "participant_name:university_id:encrypted_password:SID".
      viii. Unlock the password file.
        ix. If the personal information file already exists show an error message and leave
            the script.
         x. Save the personal information in its file, inside the participant's folder.
        xi. Save this page's VIS in the drop-out file.
       xii. Send the password to the participant in the registered e-mail address.
      xiii. Show a final message, saying that the password was sent through mail, giving
            a link to the login page.
   (d) Else:
         i. Show an error message.
        ii. Show again the registration form, with the previous data (with the value for
            SID and the updated VIS value being sent as hidden fields).
       iii. Highlight the problematic fields.
```

Figure 3: Algorithm for regp.php3 – the registration form.

```
1. Get the PID for this process.
2. Get VIS and SID through POST.
3. If this is not the submission:
   (a) If it was not called from login.php3:
       i. Show error message.
   (b) Else:
       i. If this page has already been called from login.php3 show an error message
          and stop the script.
       ii. Update the drop out file with the value of VIS.
       iii. Randomly choose an experimental condition (see Figure 5).
       iv. Save this condition in the personal information file.
       v. Create a file to flag that the participant has passed through this page.
       vi. Set VIS = 1, to indicate that this is the first dialogue.
       vii. Calculate the maximum number of words allowed in this summary.
       viii. Show instructions according to the viewpoint and summary length.
       ix. Show the first dialogue in the sequence and the summary form (sending VIS,
           SID and the experimental condition as hidden fields, along with the dialogue
           summary, through POST).
4. Else:
   (a) Get the summary and experimental condition through POST.
   (b) If this summary has already been submitted, show an error message and stop the
       script.
   (c) Calculate the maximum number of words for this summary.
   (d) If the summary is blank:
       i. Show an error message.
       ii. Show the dialogue and summary form again.
   (e) Else:
       i. If the summary has exceeded the maximum allowed length:
          • Show an error message.
          • Show the instructions again, according to the point of view and summary
            length.
          • Show the dialogue and summary form again.
       ii. Else:
          • Save the summary in its corresponding file.
          • Save drop-out information (VIS) in its file.
          • Increment VIS.
          • If VIS < 5 (it was not the last dialogue):
            – Calculate the maximum number of words for the next dialogue.
            – Show the next dialogue, its instructions and summary form.
          • Else:
            – Change VIS to reflect that the next page belongs to the "Classification"
              stage.
            – Show the instructions for the classification of the dialogues.
            – Show the dialogues again along with their classification form.
```

Figure 4: Algorithm for pagd.php3 – the actual experiment.

to be summarised, in the order defined by the given experimental condition. Dialogues were shown one by one. Participants saw only one dialogue and its empty summary form per submission, giving the impression that there were 4 different web pages, one per dialogue. Each dialogue also presented some scene information like, for example "In a restaurant. Dialogue between an attendant and a client.".

Participants were instructed to read each dialogue carefully and to summarise them according to the given point of view and limited to the given summary length. It is important to notice that we waited to only generate the experimental condition in this phase, after the warm-up and hard-hurdle phases.

The experimental condition codifies the point of view the summary should be built under, its maximum length and the order the dialogues should be presented to the participant. Each condition is an array of 6 digits. The first, from 1 to 3, represents the viewpoint; the second, either 0 or 1, determines whether the summary has or has not the 10% constrain in its length; and, finally, the last four digits, each one from 1 to 4, define the order the dialogues should be shown. This order followed the Latin Square distribution.

Whenever participants submitted a summary, the script would check whether they had written the summary according to the instructions and, if so, it would show the next dialogue in the sequence, along with the instructions and the summary form. Had the script found any error it would show an error message, describing the problem and presenting the summarised dialogue again, along with the submitted summary.

With the instructions, participants were informed about the number of dialogues remaining. After all dialogues were classified, the participants were presented with the dialogue classification page (see Section 3.6).

```
1. Initialise an array with all the possible experimental conditions (24).
2. Lock the conditions file.
3. Remove from the array all the conditions present in the conditions file.
4. If there is no condition left in the array:
    (a) Fill in the array with all the possible conditions.
    (b) Erase all the conditions in the conditions file.
5. Randomly choose one condition from the array.
6. Save this condition into the conditions file.
7. Unlock the conditions file.
```

Figure 5: Algorithm for the experimental condition generation.

In order to better distribute the participants into the experimental conditions, trying to get approximately the same number of participants in each condition, we used a conditions file (Figure 5). This file kept all the conditions the script had already given. So, before giving a participant a new experimental condition, the script used this file to determine which conditions remained to be given. Had all 24 possible conditions already been given, the script would clear out the conditions file and start over.

3.6 Cld.php3

This page implements the "Classification" stage in Figure 1. It shows the participants the dialogues they summarised, in the order they were summarised, asking them to classify the dialogues according to one out of five politeness categories: very impolite, impolite, neutral, polite and very polite. Figure 6 shows the algorithm for this page.

```
1. Get VIS, SID and the conditions from POST.
2. If this is the submission:
    (a) If the participant has already classified the dialogues show an error message and
        leave the script.
    (b) Get the dialogues' classification.
    (c) If there is any blank classification:
        • Show an error message.
        • Show the dialogues and classification form again, keeping the already made
          classification and highlighting the blank form. Send SID, VIS and the exper-
          imental conditions as hidden fields through POST.
    (d) Else:
        • Save the classification into its file.
        • Save VIS in the drop-out file.
        • Show the page for the questions about whether the participant had recognised
          any of the dialogues.
3. Else, show an error message (the page cannot be accessed isolated).
```

Figure 6: Algorithm for the dialogue classification page.

3.7 Rec.php3

This page asks the participant whether s/he had recognised any of the dialogues and, if so, where from. It implements the stages "Final Question" and "Final Note" in Figure 1. Figure 7 describes the algorithm followed by this script.

```
1. Get VIS and SID from POST.
2. If this is the submission:
    (a) If the participant had already answered this form:
        • Show an error message.
        • Stop the script.
    (b) Else:
        • Get the answers for the questions.
        • Save these answers in the recognition file.
        • Save the drop-out information.
        • Show a thank you note.
3. Else, show an error message.
```

Figure 7: Algorithm for the dialogue recognition page.

In this page, the submitted information was not tested, *i.e.*, the participants could choose not to answer the question. In the end of the whole process, the script showed a thank you note and an e-mail address in case the participants wanted to provide any feedback on the experiment [3].

4 Results and Analysis

Figure 8 shows the drop-out numbers for this experiment. The bars show the number of participants that visited a specific page and chose to go further. In our experiment, the warm-up and hard-hurdle techniques are represented by the bars from the presentation page ("Pres" in the Figure) to the log in page ("Log" in the Figure).

The actual experiment started in "His", *i.e.*, after the participant had seen the history page and decided to proceed in the experiment. At that moment an experimental condition was assigned to the participant, and the first dialogue in the sequence was shown.

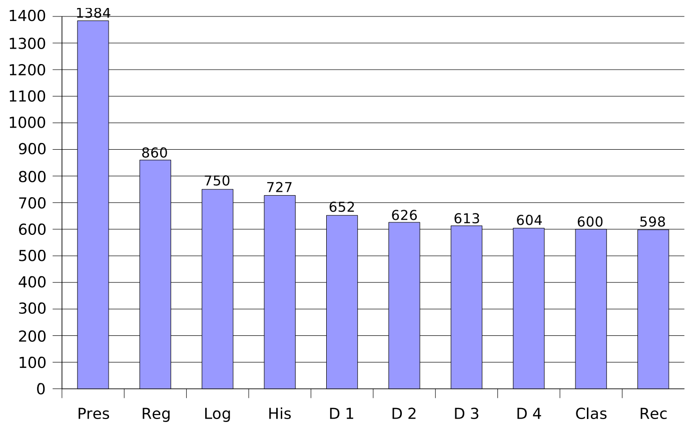

Figure 8: Drop-out levels

Like in our previous experiment [7], the Figure shows a well defined exponential curve in the number of participants who decided to move from one page to the next one. From the presentation page to the log in page we lost 657 participants, for 1384 went through the first page in the experiment and 727 saw the first dialogue. In the experimental phase, on the other hand, we lost 129 participants, considerably less. This suggests that our measures for avoiding drop-out actually worked.

Figures 9 to 11 compare the number of participants who finished the experiment and those who dropped out, according to gender, knowledge area, educational level, age and Brazilian Region where they came from. As we have these data only for those participants who actually provided them, the total amount of participants to be considered is 860.

A 3×2 table, representing the participants' gender and the number of participants who dropped out and finished the experiment, showed that the drop-out rate did not depend on the participants' gender (χ^2(2, N=860) = 4.95). The rates according to the participants' educational level, on the other hand, yielded opposite results (χ^2(2, N=860) = 14.50, at the

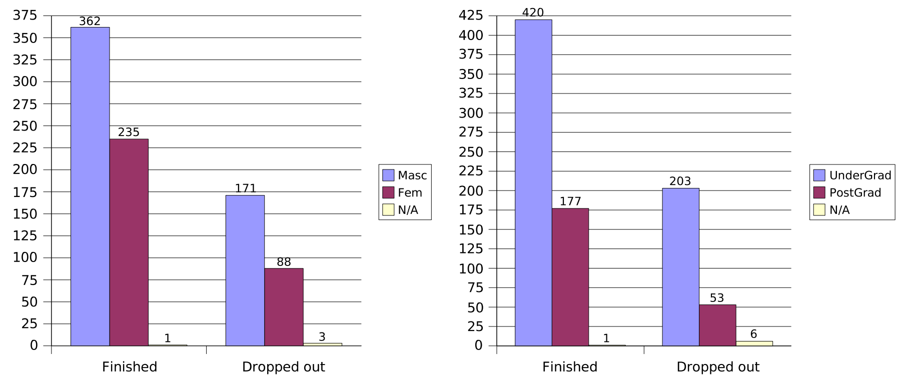

Figure 9: Number of participants, according to gender and educational level.

significance level of $p < 0.001$). Apparently, participants finishing or not the experiment depended on their educational level.

One possible explanation for this fact might be that the experiment's appeal was, apparently, different for each educational level group, being higher for those participants who are post graduate students. Another explanation might reside in the fact that, usually, post graduate students have better computers than the undergraduates, and we had some reports of black-outs happening in some undergraduate laboratories.

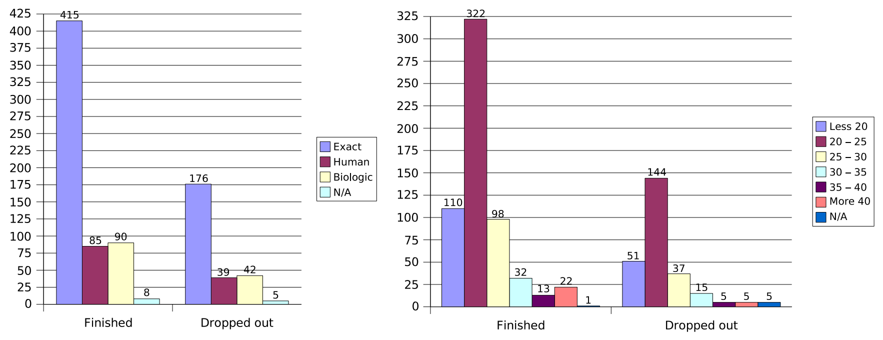

Figure 10: Number of participants, according to the knowledge area and age.

A χ^2 analysis of the data showed in Figure 10 indicates that the drop-out rates did not depend on the participants' knowledge area or age. The values for the χ^2 test are, respectively, $\chi^2(3, N{=}860) = 0.29$ and $\chi^2(6, N{=}860) = 7.47$, which is non significant. Figure 11 yields the same results, $\chi^2(7, N{=}860) = 3.86$.

Figures 12 and 13 show, respectively, the participants' distribution amongst the experimental conditions and the number of drop-outs per condition. A χ^2 analysis of both graphs altogether indicated that drop-out rates did not depend on the experimental condition ($\chi^2(23, N{=}727) = 23.09$), which otherwise might represent a risk to data validity.

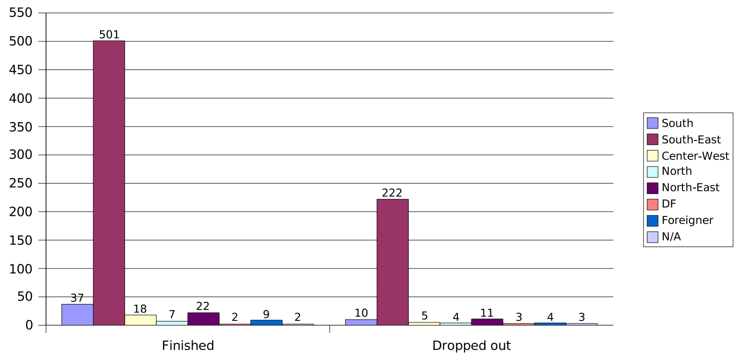

Figure 11: Number of participants, according to the Region of origin.

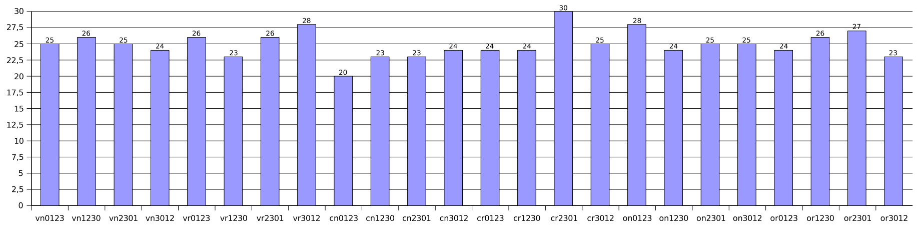

Figure 12: Participants finishing the experiment, according to the experimental condition.

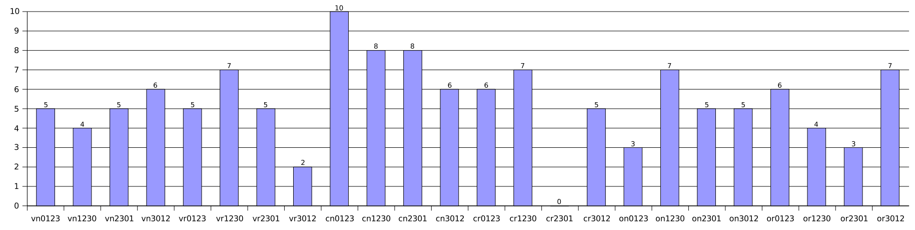

Figure 13: Participants dropping out, according to the experimental condition.

4.1 Problems and Difficulties

Despite our best efforts in trying to get the experiment to run smoothly, many problems arose, like some sporadic black-outs, which lost us some participants. Another problem we had was the amount of participants trying to access the experiment at the same time. Sometimes this number was so big that the server dropped out some participants. In all these cases we have apologised for the inconvenience and kept the dropped-out participants in the prize draw.

5 Conclusion

In this paper we described the technical details of a web experiment. Participants had to summarise four dialogues in a row. After finishing this task, they had to classify the summarised dialogues according to their politeness degree, and answer a final question about whether they had recognised any of the dialogues.

To deal with the experimental issues we followed a number of strategies from the literature (see the included bibliography), which we summarised in this paper. We also described the algorithms we used in every web page. This can give the web experimenter an idea about some technical issues like, for instance, concurrency and session management.

The strategies described in this paper seemed to work very well, confirming previous results obtained in another experiment [7] in which we used the same techniques. Our experiment developed exactly as predicted. More specifically, drop-out rates were nicely moved to a zone external to the real experiment. Participant distribution was approximately the same amongst the experimental conditions for those who finished the experiment and those who dropped out. This increased data validity, showing that our experimental conditions have not influenced drop-out, which would be a threat to data validity.

The only problem we had was the drop-out rates according to the participants' educational degree. Apparently our experiment had higher appeal to participants who were post graduate students. This might be so due to a lack of "patience" from undergraduate students; it also might be due to a lack of experience, or even expertise, from the undergraduates, for dialogue summarisation is not always an easy task. This difference might pose a problem to the experiment's generalisation, although it represents little threat to the data validity, for the educational degree was not one of the variables and, as so, it was not part of the experimental conditions.

As a final word, it is mostly advisable that someone willing to design a web experiment follows the tips in the bibliography we present here. We also hope that our description can help to address other more common technological issues that web experimenters will most certainly have to face.

Acknowledgements

We thank all the participants for their volunteer work.

References

[1] A. Frick, M. Bächtiger and U. Reips. *Financial Incentives, Personal Information and Drop-out Rate in Online Studies.* In Current Internet Science - Trends, Techniques, Results. Reips, U.; Batinic, B.; Bandilla, W.; Bosnjak, M.; Gräf, L.; Moser, K. & Werner, A. (eds.), Online Press (1999).

[2] S. D. Gosling, S. Vazire, S. Srivastava and O. P. John. *Should we trust Web-based studies? A comparative analysis of six preconceptions about Internet questionnaires.* American Psychologist, 59(2), (2004).

[3] U. Reips. *Experimenting in the World Wide Web.* In Proceedings of the 26th Society for Computers in Psychology Conference (SCiP – 96), Chicago, USA, (1996).

[4] Ulf. Reips. *Internet-based Psychological Experimenting: Five Dos and Five Don'ts.* In Social Science Computer Review, 20(3), (2002).

[5] U. Reips. *Standards for Internet-Based Experimenting.* In Experimental Psychology, 49(4), (2002).

[6] U. Reips and C. Neuhaus. *WEXTOR: A Web-Based Tool for Generating and Visualizing Experimental Designs and Procedures.* In Behavior Research Methods, Instruments, & Computers, 34(2), (2002).

[7] N. T. Roman, P. Piwek and A. M. B. R. Carvalho. *A Web-Based Experiment on Dialogue Classification.* Technical Report ITRI-04-15, Information Technology Research Institute – University of Brighton, (2004).

Appendix

Parameter	**Sender**	**Value**	**Description**
VIS	index.html	in	Signals to the next page that index.html was visited and the participant decided to proceed.
VIS	regp.php3	re	Signals that regp.php3 was visited.
VIS	login.php3	lo	Signals that login.php3 was visited.
VIS	login.php3	hi	Signals that the history page was visited and the participant decided to initiate the experiment.
VIS	pagd.php3	1	Signals that the first dialogue was summarised.
VIS	pagd.php3	2	Signals that the second dialogue was summarised.
VIS	pagd.php3	3	Signals that the third dialogue was summarised.
VIS	pagd.php3	4	Signals that the fourth dialogue was summarised.
VIS	pagd.php3	cl	Signals that the classification began.
VIS	cld.php3	fi	Signals that the final question was made.
SID	regp.php3		Contains the session id. Used to keep track of the participant across the pages.
SID	login.php3		The session id.
SID	pagd.php3		The session id.
SID	cld.php3		The session id.
MCOND	pagd.php3		The experimental condition.

Table 1: Hidden parameters sent throughout the pages.