PhD Dissertation

Epidemiology of Representations: An Empirical Approach

—original title may change—

Sébastien Lerique[1]

Supervisor: Jean-Pierre Nadal[2]
Co-supervisor: Camille Roth[3]

[1]Centre d'Analyse et de Mathématique Sociales (CAMS, UMR 8557, CNRS-EHESS, Paris). Email: sebastien.lerique@normalesup.org.

[2]CAMS, and Laboratoire de Physique Statistique (LPS, UMR 8550, CNRS-ENS-UPMC-Univ. Paris Diderot, Paris). Email: nadal@lps.ens.fr

[3]CAMS, Centre Marc Bloch (CMB, UMIFRE 14, CNRS-MAEE-HU, Berlin), and Sciences Po, médialab (Paris). Email: camille.roth@sciencespo.fr

Contents

Chapter 1

Introduction

Chapter 2

Brains Copy Paste

Chapter 3

Gistr

3.1 Introduction

3.2 Related work

- Bartlett
- Nettle
- Bebbington
- Mesoudi & Whiten
- Kirby & Tamariz
- Acerbi
- Claidière
- Smartphone psychology
- BCP
- *Missing information*: most blog and media outlet authors do not quote their source when they publish a quote online (it's often not relevant to the article), meaning there are no source-destination links in the data collected; this information must instead be inferred anew to study the evolution of content. There is also no access to author information (gender or age, experience in writing, but also psychological factors like memory span), ruling out any study of individual author effects in transforming the content.
- *Missing context*: the lack of access to the context of production and reception of quotes makes it impossible to interpret what a quotation means to its author or its reader (**???**; **???**; **???**). Analysing any kind of semantic evolution is therefore out of reach for this kind of passively collected online data (**???**, to be published).

their questions, and their limitations

3.3 Methods

3.3.1 Experiment design principles

Advantages and challenges of transmission chains

An obvious way to address the questions raised in the previous chapter is to use transmission chains in the laboratory to study the evolution of online quotations in a controlled setting: each subject reads, retains, and rewrites sentences that are then passed on to the next subject in a chain of reformulations. Such a setup can reproduce an idealised version of the read-remember-rewrite process which, we hypothesised, participates in the evolution of quotations in blogspace and media outlets. It also provides the information that our previous data set lacked in order to analyse the complete transformations of quotations, as well as the long-term effect of those changes: the links between parent and child sentences are naturally encoded in the data, such that the transformations undergone by each sentence can be studied in full detail. There is no need to restrict ourselves to simpler changes as was necessary for the inference procedure used with digital traces from blogspace. By creating an artificial setting, the experiment design also lets us control the reading and writing conditions as well as the context in which sentences appear, which further removes one of the inevitable uncertainties of the previous protocol (albeit at the cost of less ecological conditions).

The laboratory transmission chain paradigm is not a good fit for our exploratory approach however: we aim to collect data that will allow us to study both the complete set of transformations undergone by short utterances such as online quotations, and the interactions and cumulative effect of such changes; yet we do not know in advance the types of changes that subjects will make, or the extent to which such changes vary according to the type of linguistic content. Transmission chain studies typically start with an a priori hypothesis focused on a well-identified type of content, which is then empirically tested by contrasting the evolutionary outcome of two classes of sentences. Instead, our goal here is to provide first steps in characterising the process by which such evolution of linguistic content arises, and observe how it accumulates in the long term. The setup must thus allow us to collect enough data to extract regularities in successive transformations operated by different subjects on different sentences, and provide a resolving power similar to that of substitutions in online quotations so that we can compare results with the previous chapter. Since our main target is the set of detailed transformations and their interactions, a phenomenan of higher dimensionality than the contrast of accumulated outcomes, it is also crucial to fine-tune the difficulty of the read-write task and the complexity of the source sentences, in order to trigger a set of transformations varied enough that it could approach some of the changes encountered in real life situations. Our progress therefore involves a non-trivial trial-and-error component: indeed, a task made too easy or too difficult, and more so a set of source sentences that are too complex or too straightforward, will lead to either mass deletions or perfect conservation (or the former followed by the latter), none of which can help characterise the more intricate changes that linguistic content undergoes in the ecological setting we aim to simulate.

Web and smartphone experiments

Complementary to laboratory studies and to approaches using online digital traces, a new empirical approach based on Web browsers and mobile computing is striking a different balance in the trade-offs of experimental work which seems very promising in addressing the problems outlined above.

Indeed, browsers (both on desktop and mobile) and smartphones have evolved into powerful, ubiquitous application environments for which one can develop any kind of experiment involving text, graphics, and human interactions. At the cost of increased engineering requirements and a different approach to subject recruitment, Web and smartphone experiments give the designer full control over what data is collected and the way interactions are framed (similar to laboratory experiments), and make it possible to quickly collect data sets at scales comparable to what filtered and cleaned digital traces provide.

This approach makes a number of unusual trade-offs, the benefits of which can be summarised as follows:

- *Control*: similar to laboratory experiments, and unlike digital trace analysis, it is possible to use complex designs where all the interactions of the subjects are framed and observed by the experimenter. This includes for instance the presentation of the experiment (e.g. as a game or a self-improvement aid, aside from being a scientific study) and, more importantly, the ways in which the system mediates the interactions between the subjects.
- *Scale*: if and when needed, the technical platform can scale the number of subjects to the tens of thousands at low marginal cost. Interactions between subjects can also scale to involve synchronous or asynchronous contact between hundreds of people, without having to manage per-subject scheduling.
- *Speed of data collection*: once the initial development is completed (see costs below), the data collection cycle is short. One day can be enough to collect 1000–10,000 usable data points, a size comparable to the final substitutions set extracted and analysed in the previous chapter. This is especially relevant for exploratory work which is made much easier with shorter trial-and-error cycles.
- *Flexible recruitment*: while also a challenge (see costs below), subject recruitment is more flexible than in the laboratory: services like Prolific Academic[1] let the experimenter recruit at reasonable costs in pools of tens of thousands of subjects with fine-grained demographic filters. Wider audiences can be achieved by offering non-financial rewards, framing the experiment as a self-improvement application, or turning it into a game.

The corresponding costs are the following:

- *Technical challenge*: developing Web and smartphone experiments involves a substantial amount of engineering, and makes use of an array of technologies that most researchers, even technical, are not familiar with. While a couple of all-in-one kits exist,[2] creating an experiment that meets one's research questions requires learning average skills in most of the various technologies at play: a native or cross-platform smartphone development environment, Web application development, backend server programming, and some server administration skills. Most importantly, the paradigms and problems encountered are new to researchers: programming is asynchronous due to network communication and the user interface, and technicalities such as user management or email validation can grow into difficult engineering challenges.
- *Spam-control*: subjects are not constrained or encouraged by the face-to-face interaction of a laboratory experiment, neither are they (in most experiments) in the course of an interaction with friends that provides natural incentives for what they write, as can be the case with digital traces. Participants must have an incentive to perform the experiment's tasks well. If the spam introduced by one subject can be isolated in the design of the experiment, one possibility is to filter it once the data is collected and make payment depend on its prevalence. But if the

[1] https://www.prolific.ac/
[2] See e.g. http://funf.org/ and http://www.epicollect.net/.

spam introduced by one subject naturally propagates to data seen by other subjects in the experiment, as is the case for transmission chains, effective anti-spam pressures and motivations need to be factored into the design.
- *Recruitment cost*: while recruiting up to a few hundred subjects is cheaper than the equivalent for a laboratory experiment (not counting the development cost),[3] and is easy to manage for fast prototyping and pilot tests, recruitment cost rises linearly with the number of subjects and the time they spend on the experiment, unless a different strategy is used. Turning an experiment into a playful application or an application useful to the subjects (effectively making them users) involves yet another set of skills, can prove challenging, and must be factored into the development cost.

General setup for Web-based transmission chains

The balance achieved by the Web-based approach is well adapted to the experimental requirements we outlined above. Since no existing system would fit our needs, we chose to develop an in-house Web-based platform that could run all our transmission chains as Web experiments. Once ready, the platform would allow us to gather sufficient amounts of quality data in short cycles. We further decided to implement the simplest possible version of the transmission chain paradigm that is still viable, and leave the exploration of more complex setups for future research: the task we used asks subjects to read and memorise a short utterance, wait a few seconds, then rewrite what they have read as accurately as possible. We ran three main experiments using this evolving platform, and many smaller pilots in between to test improvements from the larger runs and adjust task parameters with source complexity, such that the overall quality of the data we gathered gradually increased. In what follows we present the general setup of the experiments, the data quality evaluation along with the changes implemented to improve it, and finally the adjustment of task complexity. Let us start with the architecture common to all experiments.

Subjects' productions are arranged in chains, such that what a given subject produces is used as the source utterance for the subject appearing next in a chain. In particular, the utterances to memorise are presented with no surrounding context, no distraction task is used between the reading and writing phases, and the material incentive for the task is purely monetary (although as we describe below we fine-tuned the interface to strongly encourage subjects to be conscientious). This simple setup lets us quickly gather data sets of several thousand utterance transformations, ensuring our results will be comparable to those from the set of 6177 substitutions we extracted in the previous chapter. Two parameters are left to vary: the reading time for the source utterances, computed as the number of words in an utterance multiplied by a reading factor that is to be adjusted (and may or may not be shortened by the subject), and the set of initial source utterances.

The experiment is available to subjects as a website, and passing it involves the following steps:

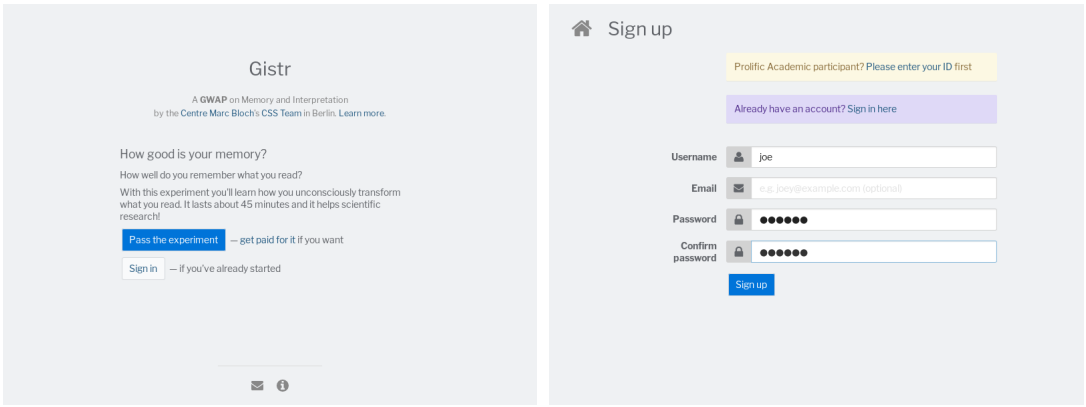
- Welcome and sign up (Figs. 3.1a and 3.1b),
- Answering a preliminary questionnaire (Fig. 3.2),[4]
- Subjects then start training for the main task, where they are asked to repeatedly memorise and rewrite short utterances as accurately as possible. As the instructions in Fig. 3.3 indicate, an utterance is presented to the subject and after a short pause they are asked to rewrite it as

[3] Global competition on online platforms like Prolific Academic drive subject payments down.

[4] An early version of the experiment also included a word span test at this stage. However, similarly to the age of subjects that we collect in the questionnaire, this data turned out to not be relevant in the analyses. The magnitude of transformations depends far more on the conscientiousness of subjects, and this non-trivial test was later removed during one of the frontend rewrites.

remembered. The process loops until the subject has completed all the utterances assigned to them (calibrated so that completing the experiment lasts at most one hour). The real trials start after 3 to 5 training trials, depending on the overall experiment length.

Each utterance from the initial set is used to create several parallel chains in order to allow for comparisons across chains with the same initial utterance. The final data thus consists in a set of reformulation trees, where each tree branch is a transmission chain started from the tree root, and continuing until it reaches a target depth defined for the experiment.[5] The number of branches in a tree is also adjusted for each run of the experiment. Except for those who drop out before finishing the experiment, all subjects are exposed exactly once to each tree in random order, such that all the reformulations in a given tree are made by distinct subjects, and nearly all subjects (excluding dropouts) are present in each tree. Satisfying this constraint means that we must always have at least as many subjects as there are reformulations in a tree. Finally, note that when exposed to a tree, subjects are always randomly assigned to the tip of one of the branches that have not yet reached the target depth: subjects are thus randomly distributed across branches, but their depth-ordering loosely corresponds to time of arrival on the tree. In particular, if a subject starts the experiment after most other subjects have completed it, he or she will be mostly exposed to utterances deep in the branches. Due to the chained nature of the data, there is no economical way of countering this ordering bias.[6] Fig. 3.4 shows a representation of the shape of the final trees.



(a) Welcome screen (b) Signup form

Figure 3.1: Initial steps for a subject entering the experiment.

Technically, the platform is a complete Web application based on current technologies, with accompanying backend server to collect and distribute utterances.[7] The experiment is available at

[5] We therefore use the terms "chain" and "branch" interchangeably in what follows.

[6] The following three approaches could be combined to counter ordering bias. (1) Have each subject do a single trial, that is, use as many subjects as there are reformulations in the full experiment; this is extremely expensive as there is a fixed minimal price for each subject, in order to give time to explore the interface, answer the initial questionnaire, and train for the task. (2) Have each subject wait an adjustable amount of time between each trial, to open the possibility of ordering subjects differently from their time of arrival; this is also expensive, as it means paying subjects for waiting most of the time they spend on the experiment. (3) Optimise the order of tree presentations of each subject so as to spread subjects across depths; while this approach could achieve some level of spread when combined with (2), it is contingent on the starting times of subjects and their synchronisation, which we do not control (subjects find the experiment through Prolific Academic notifications and are free to start whenever they want).

[7] The frontend first used the Ember.js framework (Ember.js contributors 2017), and was later rewritten and extended using the Elm programming language (Czaplicki and Elm contributors 2017). Indeed, the assurance of no runtime exceptions that

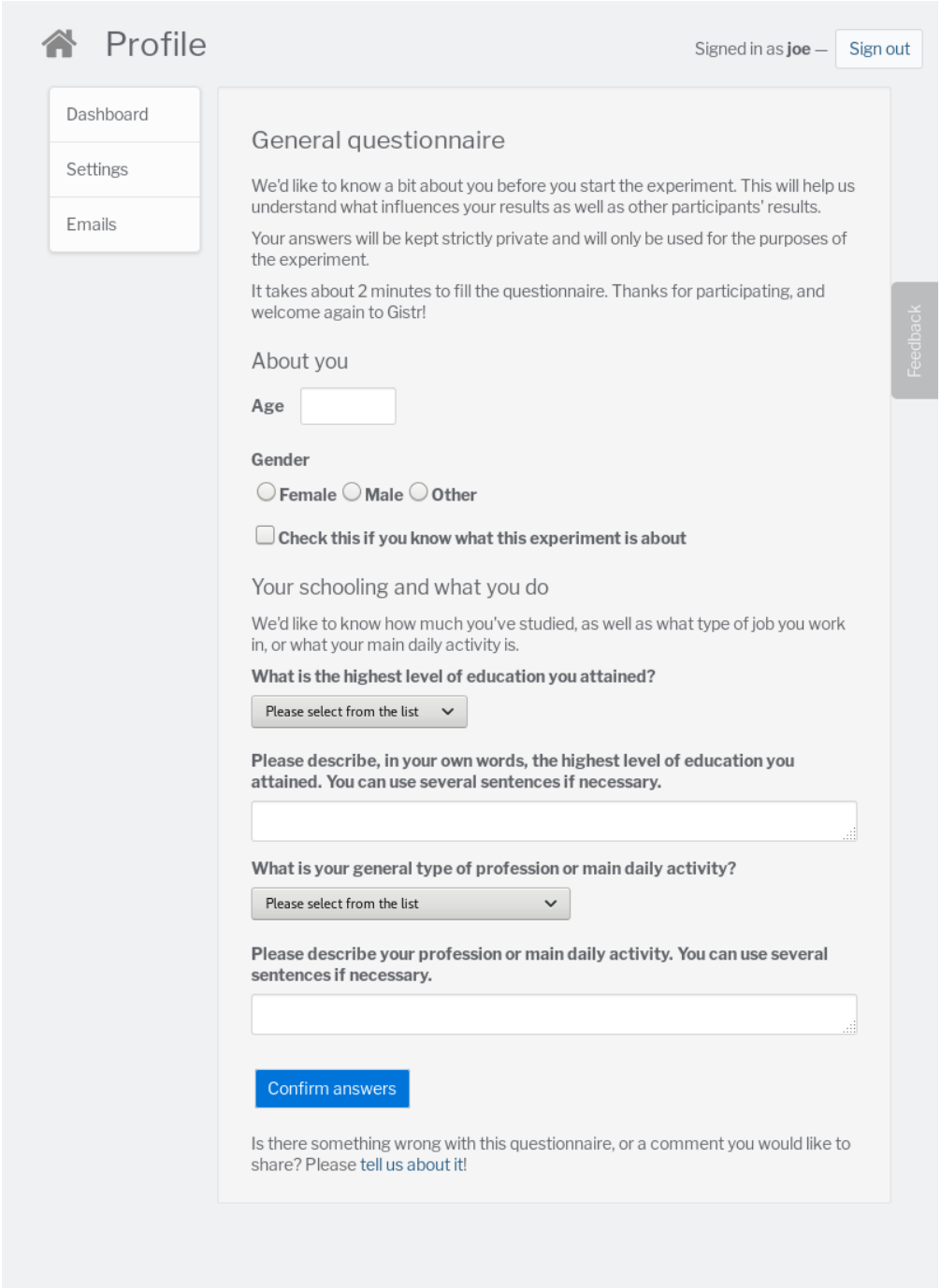


Figure 3.2: Initial questionnaire. Subjects can additionally submit feedback on the questionnaire or any other aspect of the experiment on most screens of the website.

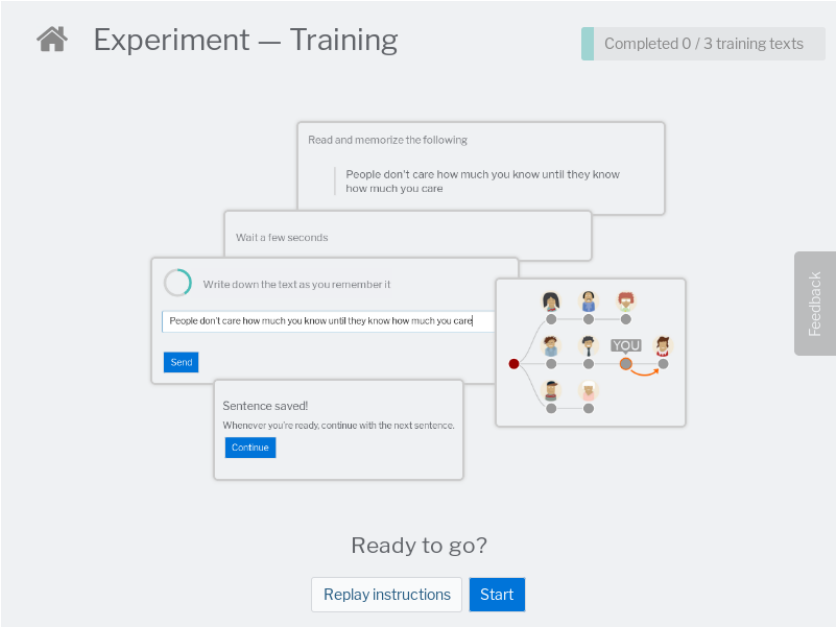


Figure 3.3: Instructions for the main task

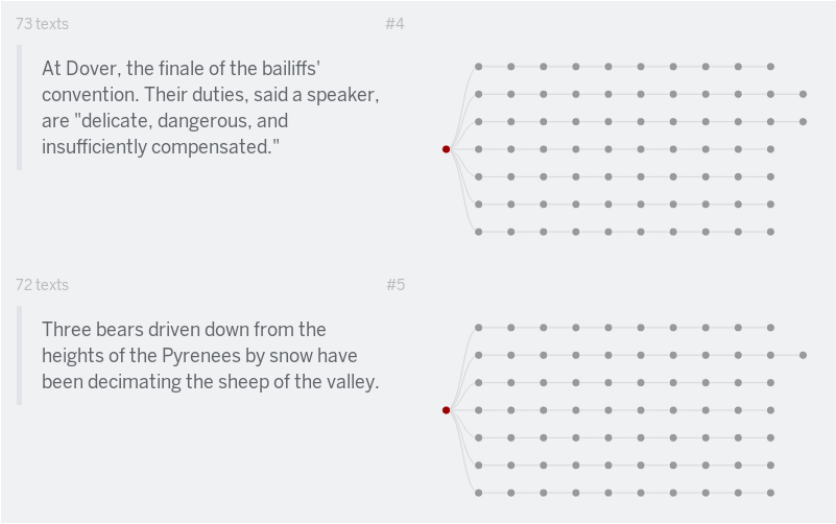


Figure 3.4: Two example reformulation trees generated by the setup, targeted at 7 branches of depth 10: the text on the left is the initial utterance for all branches of a tree, represented by a red dot in the right-hand graph; each grey dot in the graph represents an utterance produced by a subject on the basis of the preceding dot. Most subjects created one reformulation in each tree; however, since subjects from Prolific Academic do not always complete the whole set of utterances assigned to them, we recruit additional subjects to fill the trees that were left incomplete. This leads the other, already complete trees to receive more reformulations than needed, making some of their branches run deeper than the target depth (as is the case for the trees shown here). All branches are cropped to the target depth for analysis.

https://gistr.io and subject recruitment was done using Prolific Academic, a service analogous to Amazon Mechanical Turk and geared towards academic research.[8]

Using the Prolific Academic service allowed us to select among a pool of over $26{,}000$ subjects, for which we used the following criteria:

- First language English speaker
- At least 18 years old
- Current country of residence and place of most time spent before turning 18 must both be the UK
- Normal or corrected to normal vision
- No diagnosed literary difficulties
- Completed secondary school
- Not having participated in any of the preceding experiments

Only the first two constraints were enforced for the first experiment, and the full set was used for all subsequent runs. The full filter provided over 2300 eligible subjects, from which the service automatically sampled the number of subjects we requested.

Experiment 1 was the first non-trivial launch of the platform, with an initial 48 subjects, 54 root utterances, and trees targeted for 6 branches of depth 8. Subjects took an average 64 minutes to complete the experiment, and were rewarded with £6.5. A software bug that appeared and had to be fixed halfway through the experiment led the Prolific Academic service to recruit more subjects than was originally asked for, and the final number of participants was 53,[9] gathering a total 2695 utterance reformulations (above the planned $54 \times 6 \times 8 = 2592$ reformulations). Manual inspection showed that large portions of the data were of poor quality, both linguistically and because of technical inefficiencies leading to badly shaped trees; the sections below provide further details on these questions. Pilots following Experiment 1 were therefore aimed at improving data quality and solving tree shaping issues. Experiment 2 was launched with 49 subjects, 50 root utterances, and trees targeted for 7 branches of depth 7, gathering a total 2450 utterance transformations. Subjects took an average 43 minutes to complete the experiment, and were rewarded with £6. Quality issues in this data set were solved, but the choice of source utterances proved too easy to trigger varied transformations. After pilots exploring different fits of task parameters with source complexity, Experiment 3 took advantage of a more complex set of source utterances. It was launched with two batches of 70 subjects each receiving 25 root utterances, and trees targeted for 7 branches of depth 10, gathering a total 3546 utterances transformations. Subjects took an average 37 minutes to complete the 25 transformations, and were rewarded with £4.25 on average.

We now detail the evaluation of data quality and the measures that were taken to improve it. The section after this will focus on the fit of task parameters and source complexity, before moving on to the analysis and results.

Elm provides was a strong argument in favour of switching, as was made clear by the trying "customer support" experience of a bug hitting 40 to 50 subjects at once during Experiment 1. The backend is a Python application written on top of the Django REST framework (Christie and Django REST framewor contributors 2017). Most of the critical logic in the software is verified using automated tests, and the full source code is available under a Free Software licence at https://github.com/interpretation-experiment/gistr-app (frontend), and https://github.com/interpretation-experiment/spreadr (backend).

[8] The public url of the experiment was not advertised anywhere else, and checking the subjects' Prolific Academic ID confirmed that only people from that platform participated in each experiment.

[9] The bug appeared only once a large proportion of trees had reached their target depth, and then affected all the subjects nearing completion of the experiment. The time taken to respond to complaints and realise that the experiment had to be paused led some subjects to exceed the maximum allowed time on Prolific Academic, and the service then sent the experiment out to new subjects. After fixing the bug, most subjects who had started the experiment accepted to finish it, leading the final subject count to be higher than originally requested.

3.3.2 Data quality

The choice of a Web-based setup sets the requirements of the interface much higher than for a laboratory experiment. There is no opportunity for a face-to-face walk-through of the experiment or for questions, and subtle changes in the way the interface reacts to actions can lead subjects to interpret a signal where none was intended, or conversely to not notice an important message. The time of the subjects is not booked, and not having to travel to the laboratory or to talk to someone renders the interaction free of any commitment and generally more consumable: subjects can leave whenever they want, without having to feel bad about it (the only cost being the loss of their reward). The lack of human interaction with the experimenter also removes a natural incentive for subjects to take their time and perform according to what the experimenter in front of them explained. Combined together, these factors mean that if the interface is strenuous or ambiguous in any way, subjects will often pick the interpretations that make the process faster and either complete the experiment with minimal engagement or drop out. Redacting detailed textual instructions often makes matters worse. Instead, the interface must lead the subjects through the necessary explanations while remaining enjoyable, and must be unambiguous while still hinting towards the expected behaviour at the right moment, either through subtle interface reactions or through explicit contextual aids.

Manual spam-coding

Failure to properly encourage and wherever possible enforce the experiment's expectations led to data riddled with spurious transformations. Indeed, manual inspection of the data collected through Experiment 1, for which a substantial effort on instructions and for the overall interface had already been made, showed that large portions of the data was not usable as such. We therefore spam-coded all the utterances from this and subsequent experiments by hand. An utterance with any of the following properties was coded as spam:

- An ellipsis ("…") or other special characters (e.g. ">", "<") are present
- The utterance is partly or completely addressed to the experimenter (e.g. "Sorry, I can't remember")
- Over half the words are misspelled
- The utterance has no relationship to its parent utterance (i.e. it is an entirely new utterance)
- The utterance doesn't stand as an autonomous sentence, either because it is truncated or because so many words are garbled it becomes nonsense

Note that the last two criteria are not sharp, and several borderline cases had to be decided for the last one in particular. In Experiment 1 for instance, a subtle misunderstanding allowed by the interface led subjects to submit some sentences truncated at exactly 10 words, without regard to their meaning (see the details below); such utterances were unambiguously incomplete, and were thus coded as spam. In subsequent experiments however, utterances that could be made complete with the addition or the deletion of a single, sometimes unimportant word, were questioned by the same criterion. For instance the simple sentence "Mr Jones was robbed during" can be completed by adding the word "dinner" at the end, or by removing the word "during". Such sentences do not seem tied to a misunderstanding of the task, and are arguably attributable to temporary distraction whose effects are relevant to our analysis. The benefit of the doubt was given to such utterances, and they were not coded as spam.

Spam in transmission chains has the additional property of invalidating all the utterances that are made after it, such that the total number of utterances to discard is more than the spam introduced by subjects. Coded this way, Experiment 1 showed an accumulated spam rate of 22.4%. Combined with

an initial technical oversight that led a small portion of utterances to be misplaced in the chains,[10] a total of 25.9% of the utterances generated by Experiment 1 were discarded. Apart from reducing the size of the usable data, spam also leads to uneven chains across trees, a heterogeneity that complicates the analysis. Accepting this level of spam was therefore not an option.

The main tool we used to reduce the level of spam is the user interface. As explained above, minor changes in the way the interface reacts to the subjects' actions combined with relevant context-dependent information can have a comparatively large impact on the spam rate.

User interface improvements

The situation is similar to that of surveys, where much effort is put into mitigating the risk of users engaging the minimum possible effort to complete the survey (Krosnick 2000). Successfully tuning the user interface is therefore a crucial factor in the quality of the data collected: what the interface might lead subjects to see as acceptable can easily be spam for the experimenter, and both perspectives must be aligned as much as possible. Interface design problems appeared repeatedly throughout the development of the platform and the pilots. The most important points can be summed up as follows:

- *Preventing digital copy-paste*: an obvious workaround to the task that most subjects will try in the first few trials.
- *Constraining the input*: a well-known behaviour in transmission chains of linguistic content is the rapid reduction in size of the content that is transmitted.[Citation needed] In order to encourage subjects to rely on what they remember, and prevent them from quickly reaching empty sentences, an early version of the experiment would disable the "send" button if the subject's input was shorter than 10 words (Experiments 2 and 3 later relaxed this constraint to 5 words). However, some subjects interpreted the button becoming active after 10 words as a signal that their input was ready to be sent as is, even if it was only a partial sentence. This ambiguity, corrected in later versions, is responsible for a large part of the spam found in Experiment 1.
- *Improving input quality*: Experiment 1 and subsequent pilots showed the need for strong incentives to write well-edited meaningful text. Indeed when pressed for time, some subjects will tend to write misspelled, poorly punctuated, or even meaningless utterances, which invalidate all the sentences that follow in the branch. Countering this tendency involved several changes to Experiments 2 and 3: emphasis was added to the fact that subjects' productions are later sent to other subjects, encouraging a more conscientious behaviour; a bonus was associated with high-fidelity trials, and the top 5 subjects with lowest transformation rates (as defined below in the analysis) received increased payment; most importantly, input from the subjects was also checked for repeated or inadequate punctuation, and for correct spelling against a combined British and American English dictionary. Subjects were asked to correct any input that failed those tests, along with a short explanation emphasising the faulty behaviour and reminding the subject about the chain structure of the experiment. Inspecting the platform logs showed that this last measure led subjects to often correct their utterances, a fact that was also confirmed by the increased average writing time.

[10] Ensuring that no two subjects are creating reformulations for the same chain tip at the same time, while not blocking other subjects from moving on with the experiment, is a non-trivial technical hurdle. Not solving it leads the chains to have "forks", that is, utterances with several children (possibly sub-branches) instead of a single one. One of the children must then be chosen to form the main chain, and the others discarded. Solutions to the problem are difficult to test in practice, as they involve simulating dozens of subjects concurrently sending utterances to the platform. The approach adopted in Experiment 1 relied on client-side randomisation, but proved insufficient: 3.5% of the utterances posted by subjects were forks deep in the chains. Experiments 2 and 3 relied on a mix of client-side randomisation and server-side locking to solve the problem.

- *Relaxing the time pressure*: the interface of Experiment 1 made several mistakes that worsened the inherent pressure on subjects to complete the study as fast as possible (indeed, payment on Prolific Academic is per experiment, not per time spent – which, conversely, would encourage subjects to be very slow). First, subjects could terminate the reading time of an utterance at will: while this provided a measure of the effective reading time used by subjects, it also opened the possibility of speeding through the trials. Indeed over a third of the transformations of Experiment 1 were done by using less than half the allotted reading time. This pressure was increased by the presence of a "remaining time" clock in the reading and waiting phases, similar to the green clock shown on Fig. 3.3 for the writing phase. By removing superfluous clocks, keeping the reading time fixed, and proposing a break after each utterance, Experiments 2 and 3 relaxed the time pressure on the subjects and improved the final data quality.
- *Feedback channel*: survey design handbooks regularly insist on the importance of providing a channel for subjects to comment on the questions they were asked, and encourage the use of debriefing sessions to deepen that understanding (de Leeuw, Hox, and Dillman 2008). Such feedback channels have also become a norm in online services, and we therefore chose to give the possibility for subjects to comment on most screens of the experiment (excluding the read-write screens) through a side-ribbon which, when clicked, would overlay a comment box (see Fig. 3.5). It seems, however, that a more interactive option would be more effective, as only a handful of subjects entered comments over the course of Experiments 2 and 3.
- *Instructions*: finally, a continuous effort was invested into fine-tuning the exact phrasing of instructions, and making the interface for instructions palatable using the now common pattern of highlighting and surrounding an area, and adding a tooltip with short instructions next to it.[11] Here too, Experiment 1 and subsequent pilots allowed users to skip these instructions, leading a portion of the subjects to effectively never read them. Experiments 2 and 3 made navigating the complete list of instructions mandatory in order to start the trials.

These changes reduced the spam rate drastically. On the same criteria as Experiment 1, Experiment 2 showed an accumulated spam rate of .8%, which combined with misplaced utterances led to a total 1.4% of utterances discarded. Experiment 3 showed an accumulated spam rate of 1.0%, and with misplaced utterances had to discard a total 1.1% of the data.

3.3.3 Task difficulty and source complexity fit

3.4 Results

3.5 Discussion

The need for embedding

Any quantitative study relies on abstracting out details of particular cases by reducing (most often averaging) values in each dimension to a few indicators. Being able to render a precise view of the studied phenomenon then depends on being able to determine which are the right dimensions to describe it, and having access to them (**???**).

Embedding experiments in the everyday life of subjects gives access to dimensions that can be otherwise unavailable: through the use of smartphones, an experiment designer can trigger interactions

[11] The pattern was popularised by software libraries such as Intro.js (http://introjs.com/).

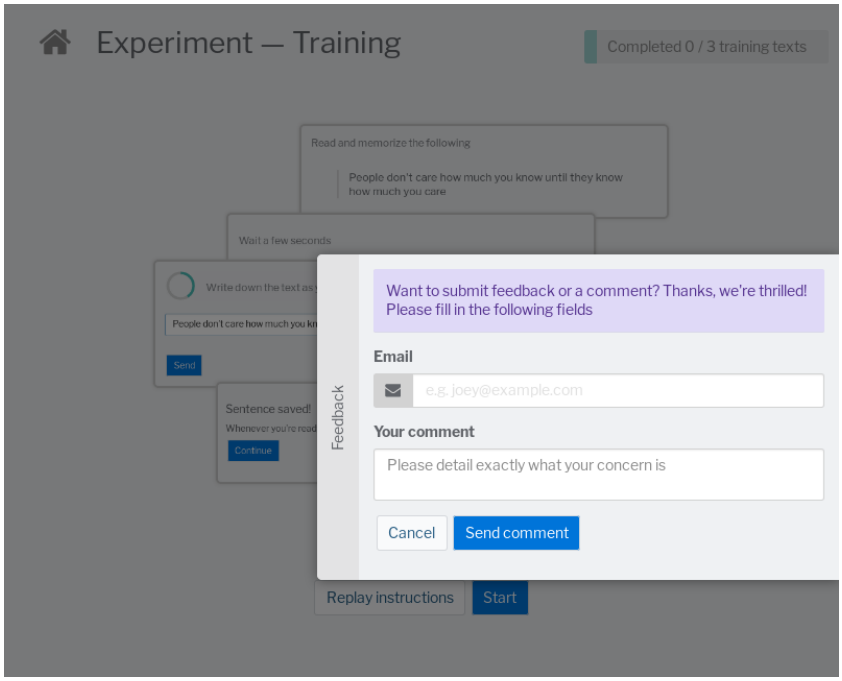


Figure 3.5: Overlay feedback box opened in the instructions screen from Fig. 3.3. The box is available in most screens of Experiments 2 and 3.

with subjects (for instance asking questions) at any moment of the day, or have measures running while subjects are offline (using the ever-increasing number of sensors present on the devices), both of which are impossible with digital traces. Above all, embedding an experiment means getting greater access to context, which opens the possibility of understanding phenomena the way they are meant in the lives of subjects, and not only in the way they are construed by the experiment designer.

Daydreaming is an example experiment developed as a smartphone application with Vincent Adam, Mikaël Bastian, Jérôme Sackur, and Gislain Delaire, that took advantage of this embedding. The experiment, focused on our awareness of daily mind-wandering, would probe subjects during a month at random moments of the day to ask them if they were mind-wandering (and, if so, what were the qualities of their thoughts).[12] While our team spent over a year developing the application, it allowed asking questions related to ecological situations which cannot exist in laboratory or passive collection studies. section 3.5 shows a sample question asked to the subjects, and section 3.5 shows an example of the results produced at the end of the experiment (this particular screen shows the results for one subject; seeing their own results was part of the reward for subjects participating in the study).

Example questionnaire in the Daydreaming app

Results on weekly mind-wandering rhythms from the Daydreaming app

- *Embedding*: as explained above, smartphone-based experiments allow for real-life embedding: the experiment designer can choose when and how interactions with the experiment and

[12] See http://daydreaming-the-app.net/ for more details.

between subjects take place, and measure any number of variables the device gives them access to (geolocation, time, phone agitation through its accelerometers, general noise level, etc.), virtually at any moment.

Chapter 4

Discussion

Chapter 5

Conclusion

References

Christie, Tom, and Django REST framewor contributors. 2017. 'Django REST Framework'. http://www.django-rest-framework.org/.

Czaplicki, Evan, and Elm contributors. 2017. 'Elm: A Delightful Language for Reliable Webapps'. http://elm-lang.org/.

Ember.js contributors. 2017. 'Ember.Js: A Framework for Creating Ambitious Web Applications.' https://emberjs.com/.

Krosnick, Jon A. 2000. 'The Threat of Satisficing in Surveys: The Shortcuts Respondents Take in Answering Questions'. *Survey Methods Newsletter* 20 (1): 4–8.

Leeuw, Edith D. de, Joop J. Hox, and Don A. Dillman. 2008. *International Handbook of Survey Methodology*. New York, NY, USA ; London, UK: Taylor & Francis.