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

TBD

3.2 Related work

Inspired by the selectionist models of culture developed by Boyd and Richerson (1985) and Cavalli-Sforza and Feldman (1981), a sizeable part of the empirical work on cultural change has focused on identifying content and context biases in the way cultural items are transmitted. This line of work relies heavily on the transmission chain paradigm initially introduced by Bartlett (1995). For linguistic content in particular, studies using that paradigm now provide a catalogue of contrasts in the way utterances or short stories are transmitted. These effects range from the stereotypical personification of objects (Bangerter 2000), the favouring of negative story aspects (Bebbington et al. 2017) or the increased hierarchical encoding of events (Mesoudi and Whiten 2004), to biases in favour of social (Mesoudi, Whiten, and Dunbar 2006) or counter-intuitive aspects of stories (Norenzayan et al. 2006; Barrett and Nyhof 2001). Other effects such as the role of emotions in the selection of items to reproduce (Heath, Bell, and Sternberg 2001; Eriksson and Coultas 2014), or conformity and prestige biases (Acerbi and Tehrani 2017) have been studied by focusing on the individual transmission step on which the evolution of content hinges.

Often, such effects are identified by selecting two or more minimally different types content and contrasting the way they evolve in transmission chains (for instance measuring the speed at which they are degraded). When a type of content is significantly better transmitted than other types, it signals that a bias is acting on that contrast dimension. The technique is useful in the context of selectionist models of culture, as it identifies examples of biases which could create selection pressures for specific cultural types and thus drive cultural evolution. In is also relevant to the more recent approach introduced by Claidière, Scott-Phillips, and Sperber (2014), which proposes the use of evolutionary causal matrices to model the change processes which are attraction-based, rather than selection-based. However, the method gives little insight into the mechanisms that give rise to such contrasts, their commonalities and differences, what they depend on, and how exactly they can be explained in terms of cognitive and situated processing.

Indeed, understanding the mechanisms behind transformations in chains, or even only quantitatively describing the details of said transformations, remains very much a challenge. This is especially true in the linguistic domain, where the complexity of language hinders most attempts to understand what is going on in a transformation. Up to now three main strategies have been developed to delve into to detail of transformations. The first is to use tightly constrained linguistic content, for instance sentences of a very specific type, or for which only pre-defined changes can happen. In that case the transformations can be directly modelled to identify regularities. The study of the recall of word lists (see Zaromb et al. 2006 for a review), and that of word replacements in short sentences (Potter and Lombardi 1990; Lombardi and Potter 1992), can be seen as employing that strategy: word lists and individual replacements in sentences are much simpler than transformations of complete sentences, and are thus more amenable to statistical analysis. Our analysis of word substitutions in the previous chapter can be categorised here too. A similar strategy is found in non-linguistic studies, such as iterated learning on sequences of colour items for which standard regularity metrics exist (Cornish, Smith, and Kirby 2013), or transmission chains of constrained visual patterns such as those used by Claidière et al. (2014). Both cases feature discrete and combinatorial pieces of content, for which it is possible to use natural notions of distance, equality, or regularity in transformation. This first strategy can be termed the "simple setting" strategy.

At the other end of the spectrum we find the "do-it-by-hand" strategy. This approach uses more ecological content but relies on exhaustively hand-coding it, and is used in most of the transmission chain studies mentioned above. The study of risk perception propagation developed by Moussaïd, Brighton, and Gaissmaier (2015), for instance, used a free-form interaction setting where subjects were taped while freely discussing a topic. The recorded conversations were later hand-coded for the presence of certain information items introduced at the beginning of the chains. The linguistic analysis of transformations of quotes in news stories provided by Lauf, Valette, and Khouas (2013) is also the product of exhaustively hand-coding differences between sentences.

Finally, the third strategy relies on pre-labelled data sets, often from online platforms, on which machine learning techniques can extract features that correlate to the transmission of pieces of content. This is the "already-coded" strategy. Danescu-Niculescu-Mizil et al. (2012), for instance, study the memorability of movie quotes by exploiting user ratings provided on the Internet Movie Database website. As we saw in the previous chapter, analysing the regularities that arise in unlabelled digital traces falls back into the first strategy, as having to infer missing information led to drastically simplifying the transformations considered.

Strategies two and three are additionally closely tied to data collection methods. Free-form interaction and more generally ecological content is costly to hand-code, and thus necessarily limited in size; it is also best used in controlled settings where the choice of content can be optimised. Conversely, using machine learning to extract features that relate to content transmission requires large amounts of pre-labelled data, which often means that an existing public data set must be used. Such studies thus seldom control the conditions under which the data is generated, which restricts the interactions they can explore to those encoded in existing data sets: any behaviour or piece of content that is not present in public data sets is off limits.

Overall, studies targeted at understanding the details of transformations of linguistic content seem forced to pick two of the following three properties, and relinquish the third: realistic content, computational analysis, and control over the generation of the data. Picking realistic content and computational analysis leads to the "already-coded" strategy. Picking realistic content and control over data-generation requires hand-coding a substantial part of the data collected, that is strategy two. Finally, computational analysis and data-generation control leads to the "simple setting" strategy. This bind thus appears as a major challenge to the better understanding of changes in linguistic content, and more broadly to the study of language-related cultural evolution. In particular, it hinders attempts to model the low-level processes which could provide a more parsimonious account of the contrasts observed in linguistic transmission chains, and allow for a deeper integration of the study of cultural evolution with linguistics.

To overcome this obstacle we turn to two related fields of research. The first, which we term the Web and Smartphone experimental approach, is creating a middle ground between controlled laboratory experiments and the analysis of online corpora. This approach takes advantage of the ubiquity of internet browsers and mobile computing to develop large-scale controlled experiments out of the laboratory. Miller (2012) discusses the possibilities opened by developing experiments as smartphone applications in particular, and notes that this method changes the logistics and context-awareness of experiments: large amounts of subjects can be recruited online without having to manage meeting schedules, and experiments can probe participants without interrupting their everyday life, both advantages that been exploited in the study of mind-wandering and happiness (Killingsworth and Gilbert 2010; Mackerron and Mourato 2013; Bastian et al. 2017). A closely related method is the development of experiments as web applications, which similarly changes the set of experimental constraints. In linguistics, the possibility for large-scale data collection has been successfully used in the study of vocabulary size (Keuleers et al. 2015; Brysbaert et al. 2016); creating studies that involve many subjects at the same time is also made much simpler by the online logistics, an advantage that has been used for instance in the study of group conversations (Niculae and Danescu-Niculescu-Mizil 2016). More generally, these approaches relax the opposition between small-scale controlled experiments in the laboratory on one side, and analyses of large-scale but passively collected online data on the other side. Once the initial development cost is covered, they make it possible to collect relatively large data sets in short cycles, and combine simplified logistics with a level of control similar to that of laboratory experiments. The second field we rely on creates an opening for the detailed modelling of utterance transformations: biological sequence alignment, the sub-field of bioinformatics which attempts to uncover commonalities in sequences of DNA, RNA, or amino acids in proteins from different species, has developed over the last 50 years a range of general algorithms to relate sequences of items. One such algorithm in particular, introduced by Needleman and Wunsch (1970), extends the principles of the Levenshtein distance and is particularly well suited to the analysis of linguistic transformations when combined with standard natural language processing methods. Inspired by Lauf, Valette, and Khouas (2013) who use similar tools to prepare their data for manual analysis, we use and extend the Needleman-Wunsch algorithm to reliably extract regularities in the way utterances are transformed through transmission chains.

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).

However, the laboratory transmission chain paradigm is not a good fit for our exploratory approach: 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 to characterise 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 phenomenon 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; it 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 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 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: control flow 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 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.

[1] https://www.prolific.ac/.
[2] See e.g. http://funf.org/ and http://www.epicollect.net/.
[3] Global competition on online platforms like Prolific Academic drives subject payments down.

General setup for Web-based transmission chains

The balance achieved by Web-based experiments is well adapted to the requirements we outlined above. Since no existing system would fit our needs, we chose to develop a tailored Web-based platform that could run 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 lessons learned in the larger runs and adjust task parameters and source complexity. The overall quality of the data we gathered thus 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.

A transmission chain is defined by a type of content transmitted, a transmission task, and a layout defining which subject production is used as the source input for the next subject. In our implementation, subjects are presented with an utterance to memorise 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). 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]
- Training for the main task, where subjects are asked to repeatedly memorise and rewrite short utterances as accurately as possible. As the instructions illustrated in Fig. 3.3 indicate, an utterance is presented to the subject and after a short pause they are asked to rewrite it as 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 started after 3 to 5 training trials, depending on the overall experiment length.

This simple setup lets us quickly gather data sets of several thousand utterance transformations, ensuring our results were be comparable to those from the set of 6177 substitutions extracted in the previous chapter. Two parameters are then 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 the set of initial source utterances.

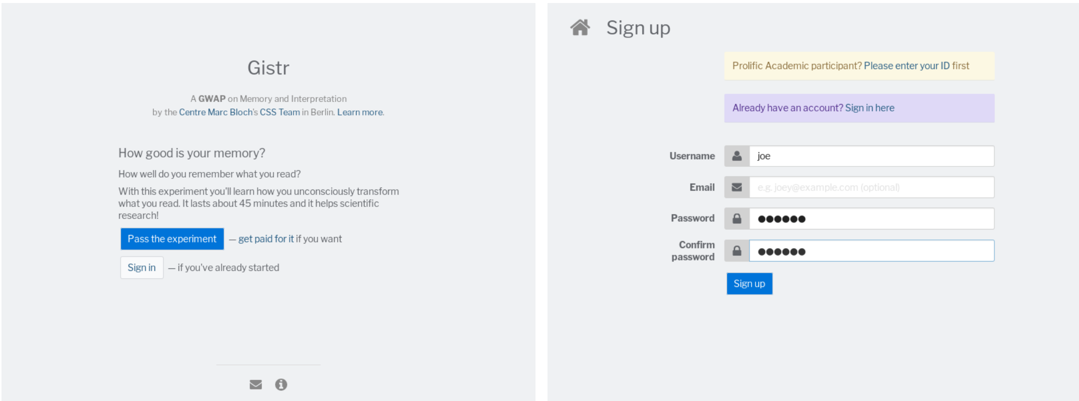
Each utterance from the initial set is used to create several parallel chains in order to allow for comparisons across chains with the same initial utterance. The final data thus consists in a set of reformulation trees, where each tree branch is a transmission chain started from the tree root, and continuing 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 dro-

[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.

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

pouts) 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. As noted, a few subjects from Prolific Academic will usually not complete the whole set of utterances assigned to them; we thus 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 a little deeper than the target depth. All branches are cropped to the target depth for analysis.

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 the 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 https://gistr.io and subject recruitment was done using Prolific Academic, a service analogous to

[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, corresponding to the time needed to explore the interface, answer the initial questionnaire, and train for the main task. (2) Have each subject wait an adjustable amount of time between each trial, to open the possibility for ordering subjects differently than 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 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 framework 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).

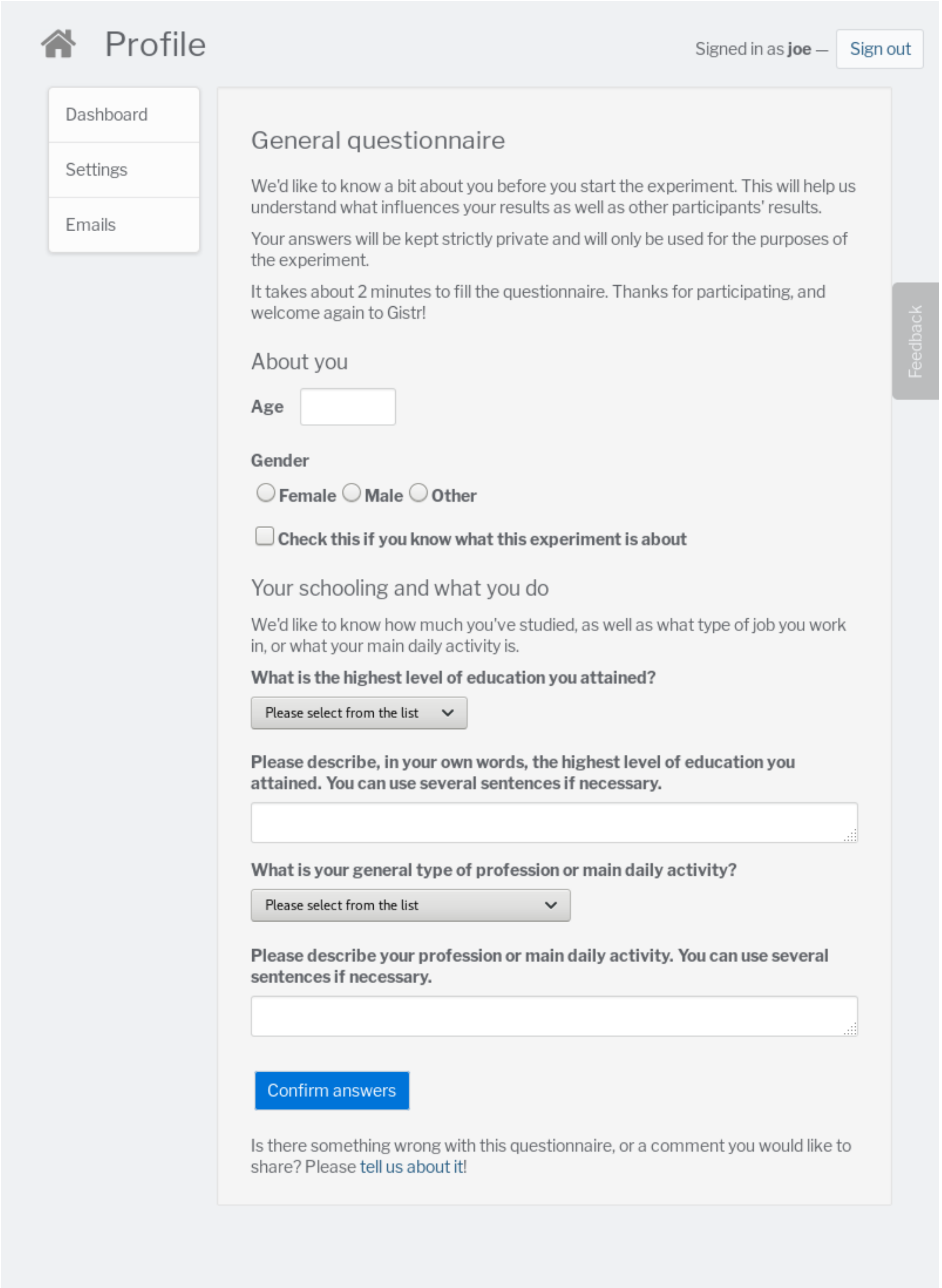


Figure 3.2: Initial questionnaire. Subjects can additionally submit feedback on the questionnaire or any other aspect of the experiment on most screens of the website.

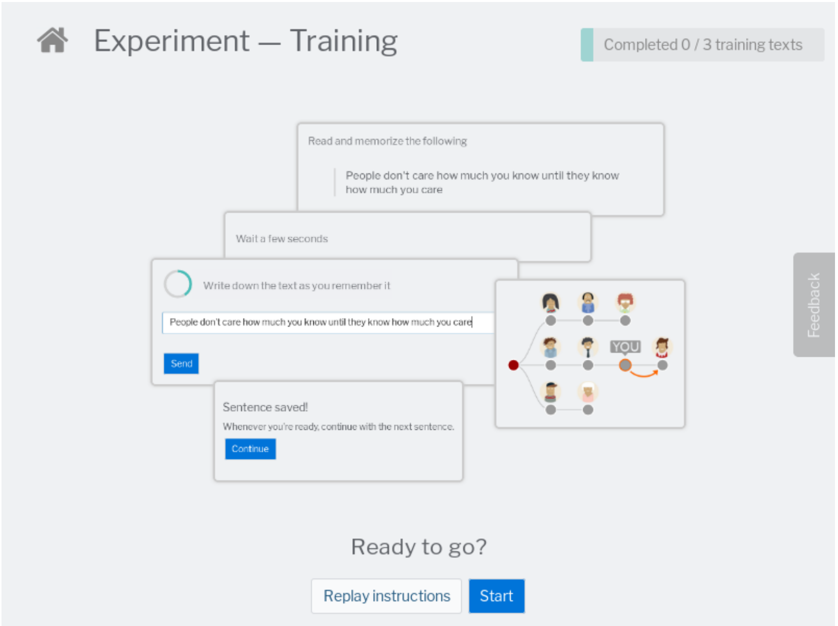


Figure 3.3: Instructions for the main task

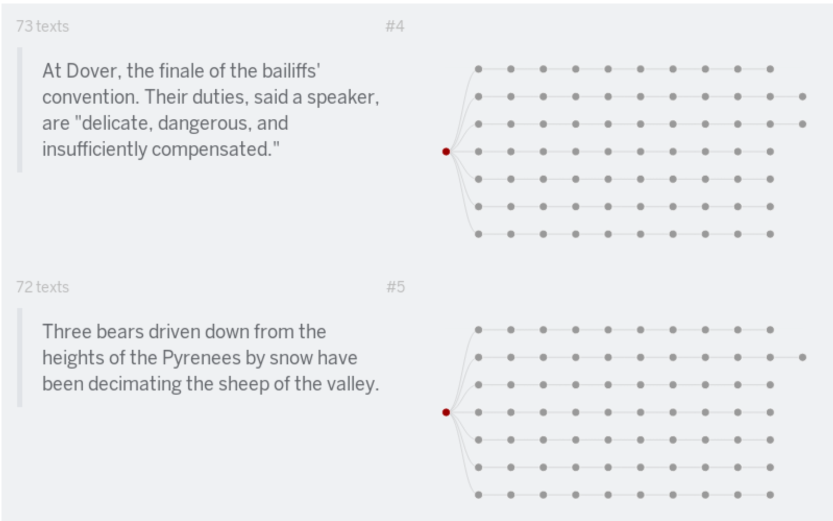


Figure 3.4: Two example reformulation trees generated by the setup, targeted at 7 branches of depth 10: the text on the left is the initial utterance for all branches of a tree, represented by a red dot in the right-hand graph; each grey dot in the graph represents an utterance produced by a subject on the basis of the preceding dot. Subjects create at most one reformulation in each tree, and most create exactly one per tree. The fact that some subjects drop out before completing all their trees leads us to recruit new subjects to fill in the missing reformulations, which is why some branches are uneven.

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 in 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 that will focus on the fit of task parameters and source complexity, before moving on to the analysis and results.

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

[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.

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. 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 were 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]

[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 extending to sub-branches) instead of a single one. One of the children must

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 (Maxwell 1936; Bangerter 2000; Mesoudi and Whiten 2004). 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 what is produced by one subject is 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. The interface asked subjects to correct any input that failed those tests, and presented them with a short explanation that emphasised the faulty behaviour and recalled 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.
- *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

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.

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 (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 a now common pattern: for the instructions pictured in Fig. 3.3 for instance, different elements or images are successively foregrounded and highlighted, and a tooltip with short explanations appears next to the active element.[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

In the simple task we used, difficulty is controlled by the reading time allotted to subjects and by the selection of source utterances. In order to unify reading times across utterances we decided that reading time would be proportional to the number of words in the utterance presented: for an utterance u, its number of words is noted $|u|_w$ and its reading time is defined as $|u|_w \cdot r$. We call r the *reading factor*. Average reading speeds for university students are usually between 200 and 300 words per minute, that is between 3.33 and 5 words per second (see Rayner, Slattery, and Bélanger 2010, where fast readers average at 330 wpm and slow readers average at 207 wpm). A reading factor of $r = 1$ therefore gives fast readers the time to read utterances more than 5 times, and slow readers about 3 to 4 times. A reading factor of $r = .3$ gives fast readers one or two readings, and slow readers at least one. $r = .2$ gives some readers one reading and others less than one, and $r = .1$ lets readers simply glance at the utterances without being able to read them completely.

Pilots and manual exploration indicated that the difficulty of the task is not linear with r. Whatever the value of r, longer utterances (more than 25 words) are often more transformed, relative to their length, than shorter utterances; longer utterances also give more space for the subjects to reformulate, leading to more changes in style and permutations in the words. Changing r has less effect on

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

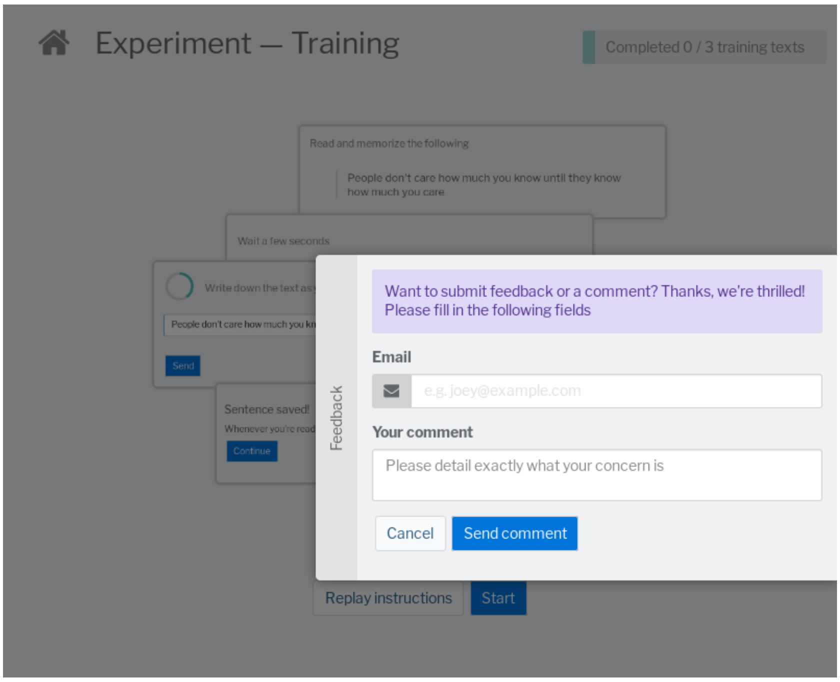


Figure 3.5: Overlay feedback box opened in the instructions screen from Fig. 3.3. The box is available in most screens of Experiments 2 and 3.

shorter utterances than on longer utterances, and on utterances in oral versus written style. For short utterances in an oral style, pilots indicated that there is an abrupt transition between a low transformation regime when subjects can read the sentence at least once, and an extremely noisy regime when the subjects do not have the time to read the utterances entirely at their normal speed. Conversely, the transition is smoother for longer utterances or utterances with a more formal written style. Choosing an adequate set of source utterances is therefore an integral factor in adjusting the difficulty of the task.

Changing the source utterances also affects the sampling bias in ways that are difficult to measure given the multidimensionality of text. Contrasting minimally different utterances in different domains has resulted in domain-specific outcomes on for instance stereotypes, information hierarchy, and counter-intuitiveness (Kashima 2000; Mesoudi, Whiten, and Dunbar 2006; Barrett and Nyhof 2001; Mesoudi and Whiten 2008), and authors have suggested that these outcomes are related to domain-specific biases in transformations (although to our knowledge these effects have not yet been studied jointly). In spite of this, we hypothesise that the low-level cognitive mechanisms underlying utterance transformation, that is the mechanisms that give rise to such accumulated outcomes, do not fundamentally change because of the type or the style of an utterance. If using news quotes instead of movie quotes or stories is likely to affect parameters of the observed transformations, it is less likely to affect the structure of the underlying cognitive mechanism, and therefore the general structure of transformations. Making this hypothesis lets us use utterance selection as an exploratory tool: by altering both the sampling of the transformations and the task difficulty, the exploration of different styles and types can help (1) improve data quality and (2) make general structure more visible, thus easier to measure and characterise. If this exploration yields insights about the structure of transformations and their effects in the long term, and if such insights are consistent with the previous chapter, then it will make sense to ask to what extent the uncovered structure is applicable to or varies with other types of utterances. Throughout pilots and experiments, our goal was therefore to find a set of utterances which would trigger varied transformations whose structure we could analyse, while at the same time helping the subjects to produce quality data by not creating too much pressure with reading time.

The set of sources used in Experiment 1 covered a broad spectrum of utterance types sampled from the following categories:

- Quotes from the MemeTracker data set used in the previous chapter,
- Famous compelling quotes from Wikisource[12] such as "Never doubt that a small group of thoughtful committed citizens can change the world, it is the only thing that ever has"
- Quotes extracted from the movie *12 Angry Men* such as "If you ask me I'd slap those tough kids down before they start any trouble, it saves a lot of time and money",
- Excerpts from news stories on controversial subjects (such as "How will the cultural and religious aspects of so many migrants impact E.U. society?") or risk-related subjects such as stories about the risks of Triclosan (used by Moussaïd, Brighton, and Gaissmaier 2015 in their study of the amplification of risk perception),
- The tale "War of the Ghosts" used by Bartlett (1995) in his original studies,[13] as well as excerpts from other tales,
- A small number of hand-crafted sentences such as surprising statements (e.g. "Don't forget to leave the door open when you leave the office") or stereotype-incongruent statements (e.g. "The young boy was suddenly hit by the little girl").

Each of these categories, we thought, could encourage the triggering of transformations. The spam level of Experiment 1, and especially the amount of misspelled words, made the exploration of the detailed transformations impossible and shifted the focus towards improving data quality through the interface. Nonetheless, it became clear that using such a heterogeneous set of utterances could surprise subjects, and was not the best approach to elicit regularities in transformations. Experiments 2 and 3 relied on a more thorough exploration of possible source data sets. Pilots explored utterances extracted from previous studies (Bangerter 2000 on personification and increased stereotypes; Heath, Bell, and Sternberg 2001 on the role of disgust; Maxwell 1936 on incoherent stories; Mesoudi, Whiten, and Dunbar 2006 for the role of social information).

Two larger and more homogeneous sets of utterances were reconstituted and finally used in Experiments 2 and 3. First, a set of movie quotes provided by Danescu-Niculescu-Mizil et al. (2012). This data set contains about 2200 pairs of quotes extracted from 1000 movie scripts; each pair is made of a quote that was marked as memorable by users of the Internet Movie Database, coupled with the closest quote in the same movie script that is spoken by the same character, has the same number of words, but is not marked as memorable on the Internet Movie Database. The 2200 pairs of quotes were filtered to keep only those which passed the spelling and punctuation quality tests from the previous section, and for which the number of words was strictly matched when excluding punctuation (this left 505 pairs). Second, a set of short stories from Fénéon and Sante (2007) was used. These stories are productions from Félix Fénéon originally anonymously published in the French newspaper *Le Matin* in 1906. They describe facts from everyday life such as accidents, suicides, or trials, in a terse and sometimes humorous style. A sample of 60 stories was extracted from the English version, for which French names and places were replaced with names and places more familiar to British subjects. Pilots explored these sets of utterances with reading factors of .1, .2, .3, .75 and 1. Finally, tests were also made using these utterances with content words replaced with pseudo-

[12] https://en.wikisource.org/.
[13] Available online at http://penta.ufrgs.br/edu/telelab/2/war-of-t.htm.

words, in order to restrict effects to the grammatical dimension only.[14] The pseudo-word tests were inconclusive, as the task became too confusing and subjects often replaced unknown words with real words.

Experiment 2 used 25 of the 27 pairs of movie quotes that had exactly 15 or 16 words, providing a homogeneous set of 50 utterances in oral style, with a reading factor of .75. Experiment 3 used 43 of the 60 short stories by Fénéon (average number of words 21.2) coupled with 4 utterances extracted from Mesoudi, Whiten, and Dunbar (2006) (average number of words 60.3) and 3 utterances extracted from the story used by Maxwell (1936) (average number of words 40.7), with a reading factor of 1.

ADD: clean counts.

ADD: mispelling proportion in exp 1

ADD: a few example sentences for each experiment

3.4 Results

Recall that the goal we set ourselves is to provide a better understanding of the process at work than what low-level feature analyses such as that of the previous chapter. In doing so we also hope to bring some light to the processes underlying high-level contrasts of utterance categories that have been extensively studied in the literature. The analysis we present is thus geared towards creating an intermediary representation of the effect of transformations on utterances, one that is at a midpoint between the low-level of word features and the high-level of category contrasts, and can be usefully modelled to better understand the evolution of utterance chains. Since this work was exploratory in nature, our presentation will also loosely follow a step-by-step development of the analysis with intermediate results. Our analysis consists in five broad steps. First, a presentation of the general trends observed in the collected data, which provide a coarse but relevant view of the behaviour of utterance reformulations in these experiments. Second, the actual procedure developed to break down transformations into smaller blocks and grasp their detail. Third, we develop a descriptive model of transformations based on the detailed view that the previous step provided. We then refine this view by quantifying the main behaviours that the model lets us identify in the transmission chains. Finally, we characterise the lexical features of the words identified by the transformation model, and show how the accumulation of transformations gradually evolves the average features of utterances. We begin with the general trends observed in the data.

3.4.1 General trends

We begin the analysis of our three data sets by examining the evolution of aggregate measures as a function of depth in the trees. Here and in what follows, the analyses are made on the data cleaned of spam, and chains truncated at their target depth: the data from Experiment 1 is truncated at depth 8, Experiment 2 at depth 7, and Experiment 3 at depth 10, and all plots are aligned to the same depth axis to facilitate comparisons. The goal of this section is to give an overview of the shape of the data and highlight a few important trends to keep in mind in the rest of the analysis. The model we develop further down will then give a more precise view of the mechanisms underlying these trends.

[14]Pseudo-words were generate using the Wuggy library (Keuleers and Brysbaert 2010).

Utterance length

A well-known effect in transmission chains with linguistic content is the quick reduction of utterance length as chains progress. These experiments are no exception: Fig. 3.6a shows a scatter plot of the number of words of an utterance $|u|_w$ versus depth in a tree.[15] The insets show the data restricted to trees for which root utterances have 30 words or less (thus most utterances in those trees also have 30 words or less); this boundary keeps all the Fénéon root utterances in Experiment 3, and we use it to separate longer from shorter utterances for the purposes of this figure. The plots confirm that word length quickly decreases as subjects read and rewrite utterances, and indicate that the reduction depends on the size of what is being transformed: very long utterances (above 100 words) are reduced to less than 100 words in 2 reformulations or less, whereas root utterances with up to 28 content words can maintain their size until the end of the branches of Experiment 3. Note that the differences in the speed of size reduction across the experiments are tied to surface features of the root utterances. Word count and average word frequency in particular, which we will later show are strongly related to transformation rate, have different distributions in the set of root utterances of each experiment: word counts are disjoint between Experiments 2 and 3, and root utterances from Experiment 2 are in an oral style, with a higher proportion of stopwords than in Experiments 1 and 3 (stopwords are alvays high-frequency words, and make up 67% of the root utterances of Experiment 2, versus 58% in Experiment 1 and 48% in Experiment 3). The steeper slopes in Experiments 1 and 3 compared to Experiment 2 are thus tied to the higher word counts and lower proportions of stopwords in their root utterances.

Next, we eliminate stopwords from the utterances and focus on the reduction in number of content words (notice however that stopword recognition is less reliable in Experiment 1 than in Experiments 2 and 3, because of spelling mistakes). For a given utterance u, the list of its content words is noted $c(u)$, and the number of content words is therefore $|c(u)|_w$, where $|\cdot|_w$ is extended to provide the length of a list of words (aside from counting words in a string). Fig. 3.6b plots the content word counts for the same utterances as those in Fig. 3.6a (in particular, insets show the same utterances in both figures), and shows a similar reduction in counts across all experiments. Here too, the differences in regression values across experiments and between the two figures are tied to the differences in distributions of word count and proportion of stopwords in the roots. In other words, the size reduction is sampled differently by each experiment: these figures show how the effect acts on each set of root utterances, but do not indicate that the mechanism is any different across experiments nor that it depends on the actual meaning of the utterances (versus primarily on surface features such as word count and average word frequency).

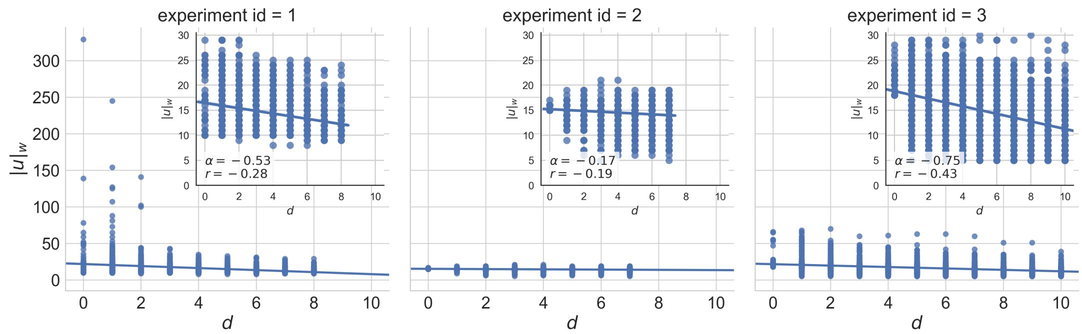
Utterance to utterance distance

As a first approximation to the magnitude of transformations we introduce a measure of the distance $\lambda(u, u')$ between two utterances u and u', defined as the Levenshtein distance[16] between the lemmas of the content words of u and u':

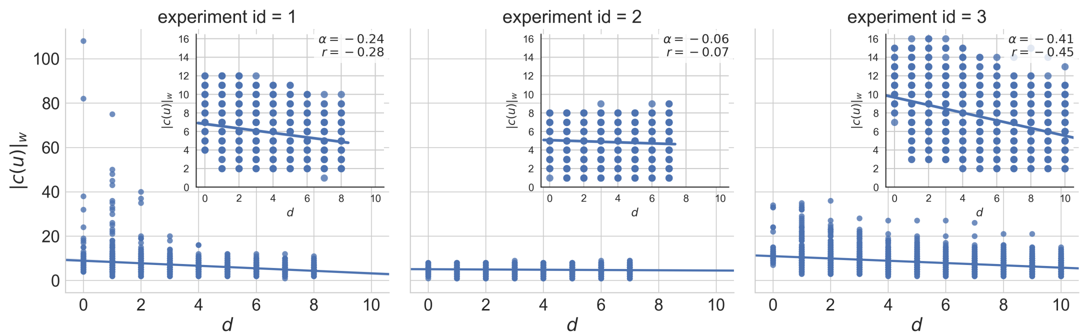
$$\lambda(u, u') = \text{lev}\left(\text{lemmatize}(c(u)), \text{lemmatize}(c(u'))\right)$$

[15] All NLP computations in this chapter are performed using the spaCy library for Python, version 1.9.0, available at http://spacy.io/.

[16] The Levenshtein distance (also known as the edit distance) is defined between two lists of items, and counts the minimal number of insertions, deletions, and replacements that are needed to transform the first list of items into the second. It has all the properties of a metric (non-negativity, identity of indiscernibles, symmetry, and subadditivity).



(a) Numbers of words, $|u|_w$



(b) Numbers of content words, $|c(u)|_w$

Figure 3.6: Reduction in utterance word count and content word count as a function of depth (d) in the three experiments. Each blue dot represents an utterance. For a given experiment, the same utterances appear in panels (a) and (b). The insets show the utterances for which the tree root has 30 or less words ($|u_{\text{root}}|_w \leq 30$), with the numerical values of the linear regression slope and correlation coefficient. All regression slopes are non-zero with $p < .001$.

For example, consider the three following utterances taken from Experiment 1 (in a tree whose root is from the MemeTracker data set):

> u_a: "This crisis did not develop overnight and it will not be solved overnight"
>
> u_b: "the crisis did not developed overnight, and it will be not solved overnight"
>
> u_c: "The crisis didn't happen today won't be solved by midnight."

After removing the punctuation and converting all words to lowercase, the lemmas of the content words of these utterances are as follows:

> lemmas$(c(u_a))$: "crisis", "develop", "overnight", "solve", "overnight"
>
> lemmas$(c(u_b))$: "crisis", "develop", "overnight", "solve", "overnight"
>
> lemmas$(c(u_c))$: "crisis", "happen", "today", "solve", "midnight"

Such that $\lambda(u_a, u_b) = 0$ and $\lambda(u_a, u_c) = \lambda(u_b, u_c) = 3$. λ thus measures differences in content lemmas and obviates minor transformations such as changes of stopwords or word inflexions. In order to have a uniform quantity across utterances of different lengths, we define the *transformation rate* ρ as the normalised distance between two utterances:

$$\rho(u, u') = \frac{\lambda(u, u')}{\max\left(|c(u)|_w, |c(u')|_w\right)}$$

ρ thus measures the magnitude of the transformation between the contents of u and u', relative to the size of the contents of those utterances. It takes its values between 0 and 1: $\rho(u, u') = 0$ if and only if u and u' have exactly the same content words in the same order, and $\rho(u, u') = 1$ means that the content words of u and u' have so little in common that rewriting from scratch is quicker than changing one into the other with word insertions, deletions, or replacements. Here, $\rho(u_a, u_b) = 0$ and $\rho(u_a, u_c) = \rho(u_b, u_c) = .6$. A major caveat of this measure is that it does not know about synonyms or expressions with similar meaning, such that two sentences separated by a transformation rate of 1 can have the same meaning at a higher level. For instance with the following sentences,

> u_d: "Will you investigate the gravest crimes of the Bush administration, including torture and warrantless wiretapping?" (from Experiment 1, originally from the MemeTracker data set)
>
> u_e: "Will you research the worst problems of the 2004 mandate, like its surveillance?" (hand-crafted for this comparison)
>
> u_f: "Don't forget to leave the door open when you leave the office" (from Experiment 1)

we have $\rho(u_d, u_e) = \rho(u_d, u_f) = 1$. The measure misrepresents the changes between these utterances, as u_d and u_e can easily be considered to have similar meanings at a high level, and their difference is far less important than the difference between u_d and u_f. Nonetheless, the measure performs reasonably well on utterances inside the same tree: in that context all utterances come from the same source and have therefore some level of meaning in common, and there is no need to differentiate between the types of transformations that $u_d \to u_e$ and $u_d \to u_f$ represent.

Transmissibility and transformation rate

Together with transformation rate, we examine a measure derived from it: the *transmissibility* of utterances, defined as the proportion of utterances at a given depth whose content lemmas are perfectly transmitted to their child, computed over all the branches of all the trees of an experiment (this measure was introduced as 'average success' in Claidière et al. 2014). If we note $\mathbb{1}_{\lambda(u,u')=0}$ the success of a subject in transmitting an utterance's content (it equals 1 if the content lemmas of u and u' match perfectly, and 0 if there was any change in content lemmas), the transmissibility $\eta(d)$ of utterances at depth d can be expressed as:

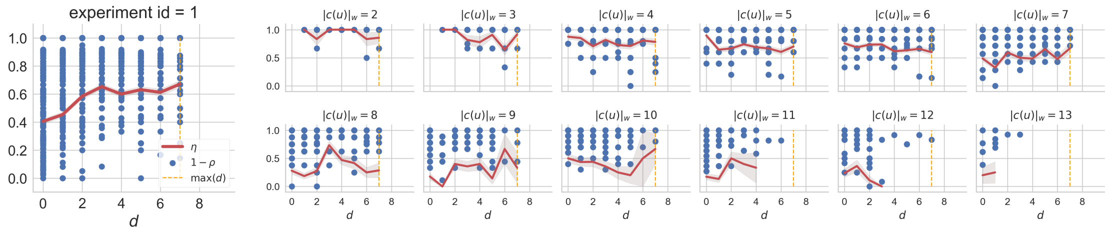
$$\eta(d) = \left\langle \mathbb{1}_{\lambda(u,u')=0} \right\rangle_{u \text{ at depth } d,\, u' \text{ child of } u}$$

Transmissibility provides a coarser measure of the evolution of transmission success than transformation rate (a change in transmissibility implies a change of transformation rates), but lets us better differentiate between the two important alternatives: perfect transmission, and transformation. A classic effect of transmission chains for various types of content is that transmissibility increases with depth in the chains.

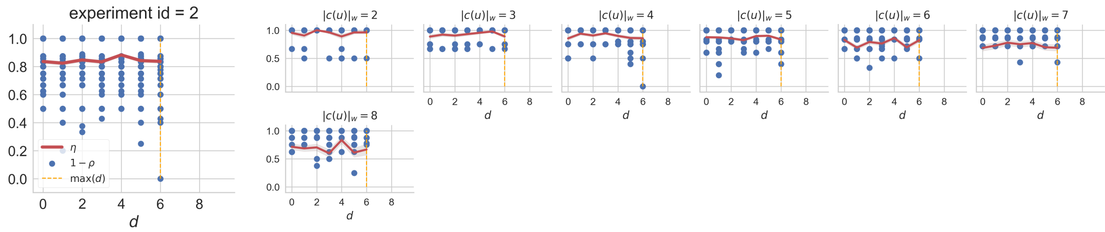
Fig. 3.7 shows the transmissibility and one minus the transformation rates ($1 - \rho$) for the three experiments, both overall and grouped by content length of the utterances. Fig. 3.7a shows an increase in transmissibility with respect to depth (from .40 to .67), when considering the whole data set from Experiment 1. However, the plots on the right show only a slight increase in transmissibility (or even no increase at all for $|c(u)|_w \notin [7, 10]$) for utterances of a given content length. The right-hand side also indicates that transmissibility depends on content length, as the transmissibility lines become gradually lower when content length increases (average .92 for 2 content words, .20 for 12 content words). Together, these trends indicate that the overall increase in transmissibility with respect to depth could be mostly due to the rarefaction of utterances with long content length: as depth increases, the proportion of shorter utterances increases; shorter utterances are better transmitted, and as consequence global transmissibility increases too.

Fig. 3.7b shows the same analysis for Experiment 2. Contrary to the previous case, transmissibility here is stable at .82-.88 with respect to depth, both for the whole data set and at fixed content length. It also depends less on content length than in Experiment 1, as utterances with 2 content words have an average transmissibility of .95, and utterances with 8 content words an average transmissibility of .69.

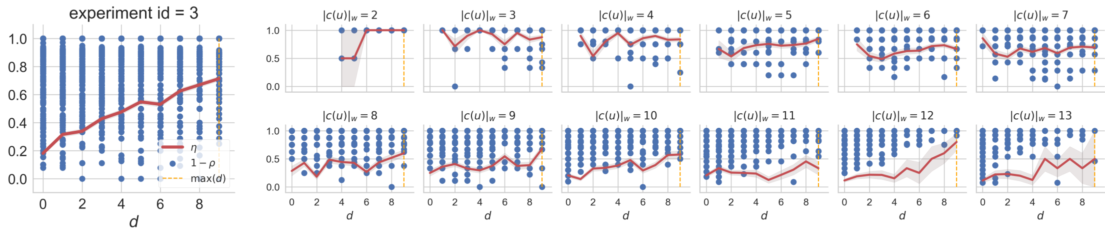
Experiment 3 (Fig. 3.7c) features an increase in transmissibility with respect to depth both globally (from .18 to .71) and at long fixed content length. This effect is stronger than in Experiment 1, and indicates that long utterances in the data set become slightly easier to transmit as they are transformed. As noted previously, utterances found at the end of a chain will often come from much longer utterances at the start, such that improved transmission success along a single branch is always mixed with the shortening of content. However, for long utterances (content lengths 8 and above), utterances found at the end of all chains are on average better transmitted than utterances *of the same content length* at the start of all chains, meaning that transmission along the chains has an effect on transmissibility of long utterances beyond the shortening of the content. Finally, the different behaviours across experiments are here again tied to the differences in word count and stopword proportion distributions in the root utterances.



(a) Experiment 1



(b) Experiment 2



(c) Experiment 3

Figure 3.7: Transmissibility and conservation rate for each experiment. Each individual graph shows both transmissibility (red line) and one minus the transformation rate (blue dots) for a subset of all utterances. Light red areas are the 95% confidence intervals for transmissibility, based on Student's t-distribution and considering each transformation as an independent event. A blue dot at $y = 1$ is an instance of perfect transmission ($\rho = 0$), and pulls transmissibility upwards; a blue dot anywhere below is a transformation instance ($\rho > 0$), and pulls transmissibility downwards. The large plot on the left shows both measures for all the utterances of an experiment. The small plots on the right show both measures for utterances that have a given number of content words (up to 13, after which the data is nonexistent or very sparse in all experiments). The orange dashed line marks the maximum depth in the experiment, so as to differentiate content lengths reaching the limit from content lengths disappearing before the limit.

Variability

We close this overview of the general trends in all experiments with a final measure: the variability of utterances at a given tree depth. For a given tree t, the variability $\kappa(t, d)$ measures the average transformation rate between all pairs of utterances at depth d in t (henceforth the *slice* of t at depth d):

$$\kappa(t, d) = \langle \rho(u, u') \rangle_{\{u,u'\} \subset \{u \text{ at depth } d \text{ in } t\}}$$

This measure gives a sense of how fast branches diverge between each other. For each experiment, Fig. 3.8 plots the variability of all slices of all trees, and the average variability averaged across trees. All three increase significantly, meaning that utterances from different branches in a tree become more and more different as the chains progress. The increase is sublinear and plateaus for Experiments 1 and 3, suggesting that branches diverge most at the beginning and less at the end. This is consistent with the increases in transmissibility. The different divergence rates correspond to the transformation rates observed in Fig. 3.7 (Experiment 2 has lower transformation rates, and diverges slower), and are therefore again tied to the differences in root utterances.

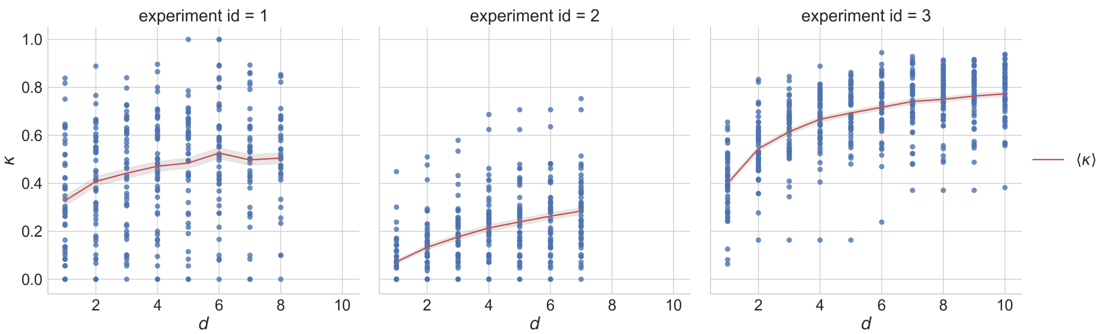


Figure 3.8: Slice variabilities in the three experiments. Each plot shows the variabilities of each slice of each tree (blue dots), as well as the average variability across slices of all trees at a given depth (red line with 95% confidence interval based on Student's t-distribution, considering each tree slice as an independent measure).

These three measures provide a coarse view of the speed at which utterance length is reduced, whether or not transformations make utterances easier to remember, and how fast the specificity of branches develops. However, they provide little insight into the detail of these trends and the transformation mechanisms that underlie them. We address this question by constructing a model of transformations in three broad steps: break down the transformations into a more detailed encoding of operations, visualise these operations to create a descriptive model of transformations, and finally quantify the main behaviours that the model allows us to observe. In what follows we focus on the data set from Experiment 3, which provides the best quality of data and sampling of transformations. The procedure we present is applicable to the other two experiments, but we will not discuss those applications here.

3.4.2 Transformation breakdown

Sequence alignments

Our first step to construct a model of transformations is to take advantage of existing generalisations of the Levenshtein distance underlying the transformation rate measure ρ. Recall that the Levenshtein distance between two sequences of items s and s' computes the minimal number of insertions, deletions, and replacements necessary to turn s into s' (and vice-versa). This problem can equally be formulated as that of aligning the items of s and s': each item of s can be paired either with an item from s' (signifying a conservation if both items match, or a replacement if the two items are different), or with nothing (signifying a deletion in the transformation of s into s'). Symmetrically, items from s' can also be paired with nothing (aside from being paired with items from s), signifying an insertion in the transformation of s into s'. In this formulation, insertions and deletions are unified into the same operation: a "gap" (or "indel" for insertion-deletion), found either in s or in s'. The problem thus formulated has become extremely important over the last 50 years in the subfield of bioinformatics known as sequence alignment.

Sequence alignment is in the business of looking for similarities between sequences of DNA, RNA, or amino acids in proteins that could indicate evolutionary or structural relationships between two or more species. Research on this problem has led to the development of several generalisations of the algorithm underlying the Levenshtein distance; these are geared towards assigning different weights or costs to the individual operations transforming one sequence into the other, finding optimal alignments of subparts of the two sequences (a task known as local alignment, in contrast to global alignment), or aligning more than two sequences simultaneously (multiple sequence alignment, in contrast to pairwise alignment).

The structure of the problem is strikingly similar to our present task: we aim to decompose the transformation of a parent utterance into a child utterance into a combination of small basic operations. In sequence alignment terms, this task is a pairwise global alignment of lists of words, for which the Needleman-Wunsch algorithm (Needleman and Wunsch 1970, henceforth NW) provides a flexible generalisation of the Levenshtein distance. For two sequences of items of any type, s and s', the NW algorithm assigns different scores to each basic operation (gap, mismatch a.k.a. replacement, and match, which is considered a scored operation like the first two), and returns the list of alignments between s and s' with maximal total score. There can be several such alignments, and each of them can be directly interpreted as a minimally scoring list of operations to transform s into s' (and vice-versa).

More precisely, let us note $s = (s_1, ..., s_n)$ and $s' = (s'_1, ..., s'_{n'})$ the items in both sequences, with n and n' the lengths of the sequences. The NW algorithm returns pairs of sequences a and a' of lengths $m \geq \max(n, n')$, made of the items from s and s' (respectively), in the same order, but interspersed with a "gap item". We noting the gap item g, and the alignment sequences $a = (a_1, ..., a_m)$ and $a' = (a'_1, ..., a'_m)$. Each tuple (a_i, a'_i) then represents the pairing of an item from s with an item from s' (either match or mismatch), or with a gap if $a'_i = g$ (and vice versa if $a_i = g$). Considered as a transformation from s to s', gap items in a' represent deletions, and gap items in a represent insertions. Each pair can thus be seen as an operation taking an item from s to construct s', and a and a' are such that the sum of scores of the operations they represent is maximal.

Take for instance the DNA sequences $s =$ AGAACT and $s' =$ GACG. An example alignment between the two sequences can be represented as follows (with the gap item represented as "-", and matches shown with vertical bars):

$$
\begin{aligned}
a &= \texttt{AGAACT-} \\
  &\ \texttt{\ \ |\ ||} \\
a' &= \texttt{-G-AC-G}
\end{aligned}
$$

The power of the NW algorithm is that gap, mismatch and match scores can be defined at compute time, knowing what items are being compared (or evaluated for a gap), and what operations would have been made up to that point if this operation were to be part of an optimal alignment. This flexibility has been used in biological sequence alignment to account for the fact that, in a DNA sequence for instance, the deletion of a base in the middle of an otherwise intact portion of DNA is less probable than the continuation of a gap that has already started. In other words, in biological sequence alignment opening a new gap is more costly than extending an existing gap, and the compute-time scores of gaps can reflect that. The same goes for mismatches: not all bases are equally likely to replace another base, and mismatch scores can factor that into the evaluation of alignments. As is hopefully clear by now, the situation is strikingly similar for sequences of words. In the next sections we detail our application and extension of the NW algorithm to decomposing utterance transformations.

Application to utterance alignment

The NW algorithm can be straightforwardly applied to sequences of any kind, provided we define scores for opening and extending gaps and a function to evaluate the comparison of two items (henceforth the match scoring function). We thus apply it to sequences of words without punctuation, with a match scoring function that takes into account the semantic distance between the two words compared. For a given pair of utterances u and u', we start by tokenising them and removing all punctuation. We then apply the NW algorithm[17] on the resulting sequences of tokens, with a match scoring function computed as an affine transformation of the similarity between two words w and w':

$$
\text{similarity}(w, w') = \begin{cases} S_C\left(\boldsymbol{w}, \boldsymbol{w}'\right) & \text{if we have word vectors for both } w \text{ and } w' \\ \delta_{\text{lemma}(w),\text{lemma}(w')} & \text{otherwise} \end{cases}
$$

where S_C is the cosine similarity function (one minus cosine distance) and $\boldsymbol{w}$ is a 300-dimensional vector representation of w encoding the word's semantics,[18] such that the $S_C(\boldsymbol{w}, \boldsymbol{w}')$ provides a measure of semantic similarity between w and w'. Finally, $\delta_{i,j}$ is Kronecker's delta which equals 1 if and only if $i = j$, and 0 otherwise. This function thus provides a "best effort" similarity measure which depends on whether we have semantic information about the words being compared or not.

Adding an affine transformation to similarity lets us adjust its importance with respect to gap scores, for which we only differentiate opening and extension scores. This definition thus uses an initial

[17] We used Biopython's implementation of the NW algorithm (Cock et al. 2009).

[18] Vector representations (also known as "word embeddings") encode words as vectors in a high-dimensional vector space. The high-dimensionality allows the vectors to bear part of the semantic information of the words as they appeared in a training corpus. Large pre-trained vocabulary sets are available in many NLP libraries, and the standard spaCy English language model includes "vectors for one million vocabulary entries, using the 300-dimensional vectors trained on the Common Crawl corpus using the GloVe algorithm" (https://alpha.spacy.io/docs/usage/word-vectors-similarities). The GloVe algorithm was introduced by Pennington, Socher, and Manning (2014), and is one of several possible methods to train such word vectors (another well-known family of methods being word2vec).

4 scalar parameters (two gap scores, two affine parameters) that define the way each operation is scored against the others. Since the final score of an alignment is computed as the sum of the scores of its individual operations, a linear scaling of all the parameters by the same amount does not change the choice of best-scoring alignments, such that we can further reduce the number of parameters by one. We choose to set the slope of the affine transformation of similarity to 1, and are then left with 3 alignment parameters:

- $\theta_{mismatch}$, the base score for the match scoring function, such that

$$\text{score}(w, w') = \text{similarity}(w, w') + \theta_{mismatch}$$

- θ_{open}, the score for opening a gap; θ_{open} is negative since it is a cost,
- θ_{extend}, the score for extending a gap; θ_{extend} is also negative.

Given the right set of parameters, the alignment produced by the NW algorithm to transform one utterance into another is a good approximation of the internal operations of said transformation. Take for instance the following two utterances from Experiment 3:

> "Finding her son, Alvin, 69, hanged, Mrs Hunt, of Brighton, was so depressed she could not cut him down."

> "Finding her son Arthur 69 hanged Mrs Brown from Brighton was so upset she could not cut him down"

TODO: use labelled lists for quotes

With the set of parameters that we obtain through training as explained below, the algorithm aligns these two utterances as follows (noting any gaps with "-", and emphasising replacements):

```
Finding her son Alvin  69 hanged Mrs Hunt of -     -    Brighton was so depressed
Finding her son Arthur 69 hanged Mrs -    -  Brown from Brighton was so upset

she could not cut him down
she could not cut him down
```

Detecting exchanges

Applying the NW algorithm in this manner works well for simple transformations such as the example above. However, more complicated transformations include operations that the algorithm does not know about. Hand inspection of the data showed that exchanging sub-parts of an utterance, in particular, is a relatively common operation for which our current tool has no representation. Consider the following two utterances from Experiment 3 for instance:

> u_a: "At Dover, the finale of the bailiffs convention, their duty said a speaker are delicate, dangerous and detailed"

> u_b: "At Dover, at a Bailiffs convention. a speaker said that their duty was to patience, and determination"

The current alignment algorithm, with parameters trained according to a procedure outlined below, produces the following:

```
At Dover the finale of the -  - bailiffs convention - -       -    -    their duty
At Dover -   -      -  -   at a Bailiffs convention a speaker said that their duty
```

```
said a speaker are delicate dangerous -   -  -        and detailed -
-    - -       -  -        -        was to patience and -         determination
```

This alignment misses the fact that the deleted part "said a speaker" is found as "a speaker said" earlier in the reformulated utterance. The general idea to detect such exchanges is that blocks of insertions and blocks of deletions can be matched against one another with the same alignment algorithm, and the resulting deep (recursive) alignment can be scored and compared to the initial shallow alignment. If the final deep score $\chi_{deep}(u_a, u_b)$ is higher than the initial shallow score $\chi_{shallow}(u_a, u_b)$, then we adopt the deep alignment with exchange as the best solution. Suppose that for the alignment of the deletion block u_- "said a speaker are delicate dangerous" with the insertion block u_+ "a speaker said that", we are able to compute an optimal deep alignment with associated score $\chi_{deep}(u_-, u_+)$; then the deep score for the top level $\chi_{deep}(u_a, u_b)$ is as follows:

$$
\begin{aligned}
\chi_{deep}(u_a, u_b) = \chi_{shallow}(u_a, u_b) \quad & \text{initial shallow score} \\
+ \theta_{exchange} \quad & \text{score the addition of an exchange operation} \\
- \text{score}(\text{deletion of } u_-) \quad & \text{recover the cost of the deletion block} \\
- \text{score}(\text{insertion of } u_+) \quad & \text{recover the cost of the insertion block} \\
+ \chi_{deep}(u_-, u_+) \quad & \text{add the deep alignment score of the exchange}
\end{aligned} \tag{3.1}
$$

where $\theta_{exchange}$ is a new negative parameter that defines the cost of creating an exchange, to be added to the existing three shallow alignment parameters. The deep alignment extension we implemented follows exactly that recursive principle, but accommodates for the possibility of multiple exchanges at each level of the recursion. Algorithm 3.1 provides an overview of the way this tree of alignments can be constructed. Note that for long utterances, the size of the deep alignment tree can grow very fast:

- For a given deep alignment, there is a list of mappings between deletion and insertion blocks,
- Each mapping is a set of (deletion block, insertion block) pairs,
- Under each such pair, there is a list of deep alignments; and from there on recursively.

Also, our implementation of the exploration of that tree is mostly brute force, and does not try to be smart in predicting which branches are dead-ends. In spite of this, we did not need to optimise the computation any further (aside from obvious gains in caching repeated computations), as most of the time of a deep alignment is spent computing shallow alignments, and most alignments of utterances are very shallow anyway. Finally, note that this approach provides no guarantee of finding the globally optimal deep alignment. Indeed, it starts from optimal shallow alignments, and explores the tree of possibles from there on. But the initial shallow alignments it extends may not be the best starting point, such that the exploration may return locally optimal deep alignments.

Nonetheless, given a good set of parameters (see the next section where we derive those), this deep alignment algorithm produces surprisingly satisfying results given the simplicity of its underlying principles. In the case of the two utterances exemplified at the beginning of this section, the algorithm produces the following deep alignment tree. First, the top-level alignment:

```
At Dover the finale of the -   - bailiffs convention |-Exchange-1------| their duty
At Dover -   -      -  -   at a Bailiffs convention a speaker said that their duty

said a speaker are delicate dangerous -   -  -        and detailed -
|-Exchange-1------------------------| was to patience and -        determination
```

Algorithm 3.1 An implementation of the deep alignment extension for detecting exchanges in NW alignments. Note that this implementation is inefficient but presentationally clearer than the implementation we made.

function SHALLOWALIGN(u, u')
    (Implemented by Biopython)
    return List of optimal shallow alignments of u and u'
end function

function MAPPINGS($a_{shallow}$)
    $\mathcal{D} \leftarrow \{d | d$ deletion block in $a_{shallow}\}$
    $\mathcal{I} \leftarrow \{i | i$ deletion block in $a_{shallow}\}$
    return $\mathcal{D}^{P(\mathcal{I})}$ ▷ $P(\Omega)$ is the power set of Ω
end function

function SCOREMAPPING($a_{shallow}$, D_M)
    $s \leftarrow 0$
    for $((u_{e,-}, u_{e,+}), D_e)$ in D_M **do**
        $s \leftarrow s + \theta_{exchange}$
        $s \leftarrow s -$ SCORE(deletion of $u_{e,-}$) $-$ SCORE(insertion of $u_{e,+}$)
        $s \leftarrow s + \max\{\chi_{deep}(a_{deep}) | a_{deep} \in D_e\}$
    end for
    return s
end function

function DEEPALIGN(u, u')
    $D \leftarrow []$ ▷ D is the list of deep alignments trees we have explored
    for $a_{shallow}$ shallow alignment in SHALLOWALIGN(u, u') **do**
        if $a_{shallow}$ has no gaps or has only gaps **then**
            $D \leftarrow (a_{shallow}, [], \chi_{shallow}(a_{shallow}))$
        else
            for M mapping in MAPPINGS($a_{shallow}$) **do**
                $D_M \leftarrow []$
                for $(u_{e,-}, u_{e,+})$ exchange in M **do**
                    $D_M \leftarrow D_M + ((u_{e,-}, u_{e,+}),$ DEEPALIGN$(u_{e,-}, u_{e,+}))$
                end for
                $D \leftarrow D + (a_{shallow}, D_M, \chi_{shallow}(a_{shallow}) +$ SCOREMAPPING$(a_{shallow}, D_M))$
            end for
        end if
    end for
    return Recursively pruned version of D with only maximally scoring deep alignments
end function

For which $\chi_{shallow} = -2.93$ and $\chi_{deep} = -2.89$. Then the alignment of Exchange 1:

```
said a speaker are delicate dangerous |E2-----|
|E2| a speaker -   -        -         said that
```

For which $\chi_{shallow} = -1.01$ and $\chi_{deep} = -0.99$. And finally the alignment of Exchange 2, from inside Exchange 1:

```
said -
said that
```

For which $\chi_{shallow} = \chi_{deep} = -0.18$.

Notice how in this deep alignment the phrase "are delicate and dangerous" was initially included in Exchange 1, only later to be recognised as a deletion in the alignment of Exchange 1. The same happened for "that", initially included in Exchange 1 and finally recognised as an insertion in the alignment of Exchange 2. Most cases of deep alignments look like this one, where a single path exists in the tree of recursive alignments. For longer utterances however, there can be several exchanges at each level, and the tree of alignments becomes much larger.

Training alignment parameters

Finally, we need to determine a set of alignment parameters that produce useful results with this procedure. Recall that the parameters are:

- $\theta_{mismatch}$, the base score for the match scoring function,
- θ_{open} and θ_{extend}, the scores for opening and extending a gap,
- $\theta_{exchange}$, the score for creating an exchange.

In order to make the problem of finding usable parameters tractable, we decided to restrict parameter training to the shallow alignment parameters only (henceforth noted $\boldsymbol{\theta} = (\theta_{mismatch}, \theta_{open}, \theta_{extend})$), and fine-tune $\theta_{exchange}$ by hand after the first three were defined (this also corresponds to the fact that deep alignments are made on the basis of optimal shallow alignments). Our general approach for this task is therefore to hand-code shallow alignments for a random set of utterance transformations in Experiment 3, then train the shallow alignment parameters to that standard before adjusting the exchange parameter by hand. Since there are only three dimensions to explore, the training step is easiest to accomplish by brute force.

We thus start by evaluating the size of the training set that is necessary to obtain a set of parameters that extrapolates well to untrained data. Indeed, a training set too small in size might provide too weak a constraint on the set of parameters, such that brute forcing would find many parameter sets that do not extrapolate well. On the other hand, manually coding alignments is time-consuming and we do not wish to code more than necessary. We used the following procedure to decide this trade-off:

1. Uniformly sample a random parameter set $\boldsymbol{\theta}^0 \in [-1, 0]^3$ and use it to generate artificial alignments for all the non-trivial transformations in Experiment 3 (i.e. for which the transformation rate ρ was positive, which amounts to 2159 transformations); note these alignments $\mathcal{A}^0$.
2. Sample a training set of size n from the artificial alignments; note the training set $\mathcal{A}^0_t$, and the remaining evaluation set $\mathcal{A}^0_e = \mathcal{A}^0 \setminus \mathcal{A}^0_t$.
3. Brute-force the sets of parameters $\hat{\boldsymbol{\theta}}_1, ..., \hat{\boldsymbol{\theta}}_m$ that best reproduce the training set $\mathcal{A}^0_t$, by exploring the sampling space $[-1, 0]^3$ with a discretisation step of .1; parameters that perfectly

reproduced the training set were always found, such that no finer-grained exploration was needed.

4. Evaluate the worst fit $\hat{f}_n$ between the evaluation alignments $\mathcal{A}_e^0$ and the alignments produced by each of the $\hat{\boldsymbol{\theta}}_1, ..., \hat{\boldsymbol{\theta}}_m$ on the same transformations.

For a given set $\mathcal{T}$ of transformations, the alignments generated by parameters $\boldsymbol{\theta}$ can be written:

$$A\left(\mathcal{T}, \boldsymbol{\theta}\right) = \{\mathrm{aln}(u, u', \boldsymbol{\theta}) | (u, u') \in \mathcal{T}\}$$

Where $\mathrm{aln}(u, u', \boldsymbol{\theta})$ is the set of alignments between u and u' produced by $\boldsymbol{\theta}$. $A\left(\mathcal{T}, \boldsymbol{\theta}\right)$ is thus a set of sets of individual shallow alignments (indeed each pair of utterances generates its own set of shallow alignments). The fit between two such sets of sets of alignments $A\left(\mathcal{T}, \boldsymbol{\theta}_1\right)$ and $A\left(\mathcal{T}, \boldsymbol{\theta}_2\right)$ is then computed as:

$$f(\mathcal{T}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \frac{1}{2} \sum_{(u,u') \in \mathcal{T}} \max_{((a_1, a'_1), (a_2, a'_2)) \in \mathrm{aln}(u, u', \boldsymbol{\theta_1}) \times \mathrm{aln}(u, u', \boldsymbol{\theta_2})} \{\mathrm{lev}(a_1, a_2) + \mathrm{lev}(a'_1, a'_2)\}$$

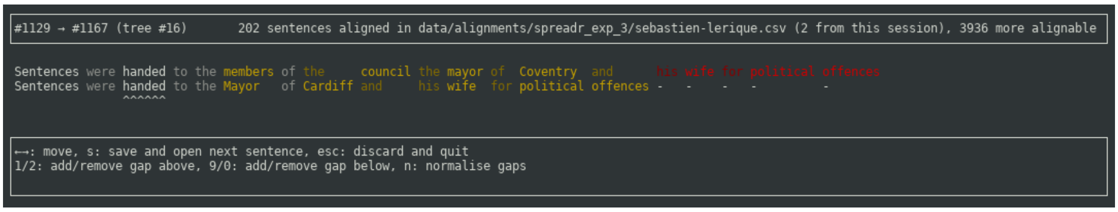
TODO: unify mathematics notations

The value of the fit thus loosely corresponds to the total number of words whose alignments would need to be changed in order to go from one set of alignments to the other. Divided by the number of transformations $|\mathcal{T}|$, it tells us the average number of word alignment errors per transformation. The worst fit $\hat{f}_n$ then gives us an upper bound estimation of the error that can be produced by training on a set of size n. One caveat in this evaluation approach is that there is no guarantee that the hand-coded alignments on which we will train could be produced by this parametrisation of alignments. We have no workaround for this caveat, other than hand-evaluation of the parameters after the training step.

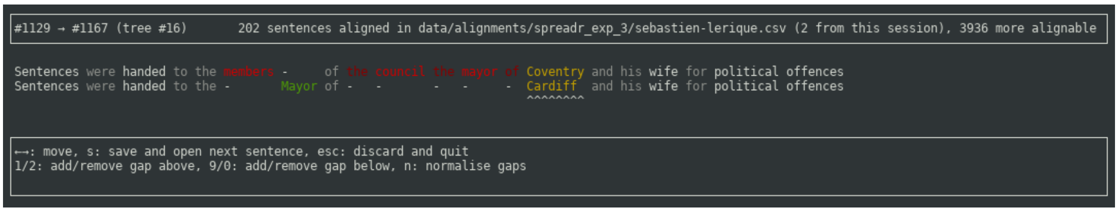
After having sampled a parameter set for step 1, we used $n = 20, 50, 100, 200$ and ran steps 2-4 ten times for each value of n. The worst values of the ten runs were $\hat{f}_{20} = 3652.5$ (1.71 errors per transformation), $\hat{f}_{50} = 1377.5$ (.65 errors per transformation), $\hat{f}_{100} = 847$ (.41 errors per transformation), and $\hat{f}_{200} = 636.5$ (.32 errors per transformation). For $n = 100$, we further resampled $\boldsymbol{\theta}^0$ ten times (step 1) and ran steps 2-4 ten times for each of those 10 parameter sets, yielding an overall $\hat{f}_{100} = 1437.5$, that is .70 errors per transformation. We conclude from this evaluation that a training set between 100 and 200 alignments is enough to reduce the final error below one word per transformation.

We thus hand-coded 200 alignments of non-trivial transformations, using a simple console interface illustrated in Fig. 3.9. The manual alignments were used as a parameter training set, on which brute forcing the best $\boldsymbol{\theta} \in [-1, 0]^3$ with discretisation step up to .025 achieved a training fit of 240 (i.e., 1.2 errors per transformation), confirming that our hand-coded alignments are most likely not possible to reproduce perfectly with this parametrisation. The final parameters obtained with this approach are $\theta_{mismatch} = -.89$, $\theta_{open} = -.29$, and $\theta_{extend} = -.12$. $\theta_{exchange}$ was then set to $-.5$ after manual trial and error.

Finally, we manually evaluated the overall quality of these parameters by hand-coding the number of errors in a random set of 100 non-trivial alignments generated by the parameters. Errors were counted as the number of words whose alignment would have to be changed in order to obtain a perfect alignment. Of those 100 alignments, 79 were perfect, 12 had 1 error, 4 had 2 errors, and the



(a) Before manual alignment



(b) After manual alignment

Figure 3.9: Console interface for manual transformation alignment. The user moves their cursor (underline below "handed" and "Cardiff") along the word sequences to insert or remove gaps and align the two utterances by hand.

remaining 5 had between 3 and 6 errors. Counting 1 error as acceptable, this method yields a successful alignment in 91% percent of the cases. To make sure this is also the case for deep alignments, we hand-coded errors in 100 random alignments for which the algorithm had explored exchanges (though not all of them are deep alignments, as it may be that the shallow alignment was the best). An error was counted for each exchange that was missing, mistaken, or should not have been present at all. Of the 100 alignments, 81 had no errors, 17 had 1, and 2 had 2 errors. The parameters obtained here were thus used for all further analyses. They are also the ones used in the example alignments discussed in the previous sections.

3.4.3 Transformation model

The alignment procedure we just presented provides a list of deep alignment trees for each transformation in the data set. At this point we have showed that such deep alignments reliably encode the details of a transformation broken down into smaller operations, thus completing the first step of our modelling approach. We now proceed to the next step: first, create an accurate visualisation of the details of transformations captured by the alignments, then derive a transformation model based on that visualisation. The step after that will then refine the model and quantify the behaviours we can measure with it.

Consensus filiations

The first step to visualise the process encoded by alignments is to reduce the possibly multiple deep alignments encoding a transformation into a single version, which we call the consensus alignment. Indeed for a given transformation $u \to u'$, a word in u could for instance be assigned to different words in u' for different choices of deep alignments. Other cases are possible, as w could be deleted in one deep alignment and not in others, and so on and so forth. A method to resolve conflicts across multiple deep alignments is therefore required.

We adopt the following procedure to construct the consensus alignment. For a given transformation $u \to u'$, start by flattening each tree of deep alignments into a list: each branch in each tree becomes a different deep alignment, and we are left with a list of uniquely defined deep alignments for the the transformation. Now for a given word w in u, determine if it is conserved either exactly or through replacement in at least half of all the deep alignments; if so, select the child word in u' which appears in most deep alignments (i.e. the majority child); if not, the consider w deleted. Any word in u' that has no assigned parent word is then considered an insertion.

A few details are worth mentioning here. First, since a word that is stable in exactly half the deep alignments and deleted in the other half is still considered stable, the procedure sometimes maintains one or two more stable words than the deep alignments it synthesises. Such conflicts are inherent to any consensus method, and the alternative in this case would be to consider the word unstable, adding deletions and insertions instead of stabilities to the consensus alignment; we choose to favour stabilities so as not to inflate operations artificially. Second, an analogous conflict can arise when a stable word has two equally probable candidate children, or conversely a given child word has two equally probable parent words. In those cases we decide in favour of the word closest to the end of the utterance # TODO: Favour the first, so as to stop suspicion of bias in position effect., and the procedure is consistent in both directions: the consensus alignment for $u \to u'$ is the same as for $u' \to u$. Finally, a small number of cases create new single-word exchanges, as two words are assigned not the same child but different children at exchanged positions; manual inspection of these cases showed that such exchanges are consistent with the transformation. In practice, only 53 of the 3461 transformations in Experiment 3 have more than one deep alignment, 46 of which have 2 (the other seven having 3 to 6 deep alignments), such that any change here has virtually no impact on the results.

With a single consensus alignment thus constructed for each transformation, it is possible to follow the ancestry and descent of individual words through parent and child transformations in a branch. Consider for instance a toy branch $u \to u' \to u''$. A word w' in the middle utterance u' is now uniquely identified either as an insertion, or as the conservation or replacement of a parent word $w \in u$ (with or without movement due to an exchange). Continuing down the branch, w' can also be linked to its child w'' (if it was not deleted), thus creating a lineage for this specific word along the branch.

Branch and utterance axes

Fig. 3.10 represents the lineages produced by this procedure on the branches of an example tree taken from Experiment 3, whose root is the following utterance:[19]

> « At Dover, the finale of the bailiffs' convention. Their duties, said a speaker, are "delicate, dangerous, and insufficiently compensated." »

At first glance the plots reflect the rapid shortening of utterances, and the fact that transformations are less important on shorter utterances deeper in the branches. The figure also indicates that word replacements, studied in the previous chapter with data from blogspace, are quite speckled: they are less frequent than deletions and insertions, and affect smaller portions of the utterances when

[19] This is tree #4, which is also shown in Fig. 3.4. The transition from depth 1 to depth 2 in branch #49 of the figure also corresponds to the example transformation discussed when introducing deep alignments (# TODO: add refs for sentences).

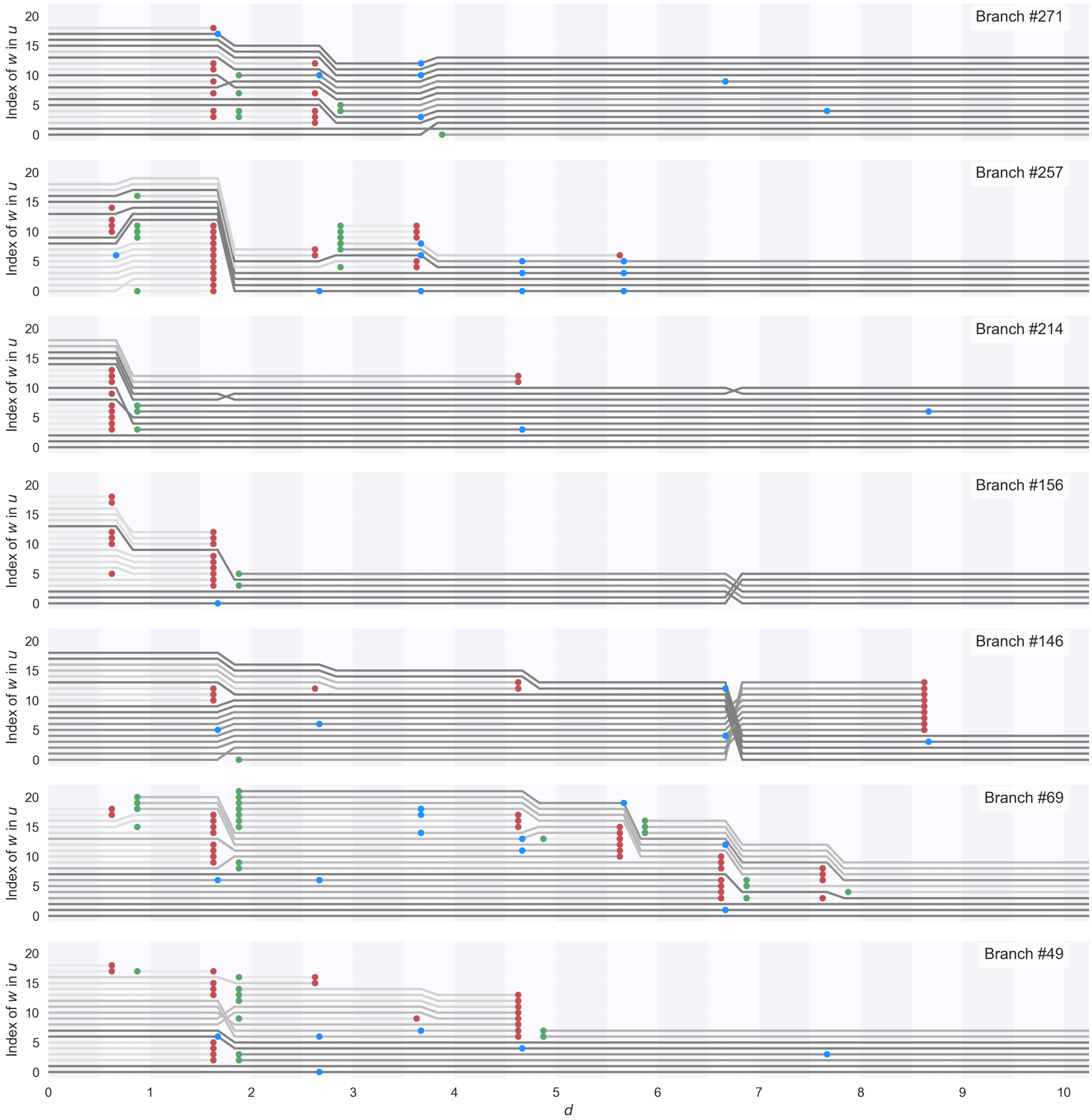


Figure 3.10: Example lineages for all the branches of tree #4 from Experiment 3. Each subplot corresponds to a different branch. The horizontal axis is the depth in the branch, and the vertical axis is the index of each word in its utterance. A grey line represents a word lineage along the branch, and the darkness of the line corresponds to the length of the path between the word's first appearance (or the branch start) and its disappearance (or the branch end); darker lines thus represent words that last longer across transformations (since branches eventually stop, however, our view of the process is truncated and the darkness is less reliable for words that appear towards the end of a branch). At each depth, the darker background band indicates what the subject sees, and the lighter band indicates the transformation that the subject made. Inside lighter bands: red dots are word deletions, green dots are word insertions, blue dots are word replacements, and exchanges can be seen when bundles of lines cross each other. Dots inside each light band are spread out on the horizontal axis so as make them easier to distinguish visually, but the horizontal position of a dot inside its band has no further meaning.

they appear. As replacements were the only process that could be extracted from the blogspace data set, suspecting this caveat was one of the motivations for our current experimental approach.

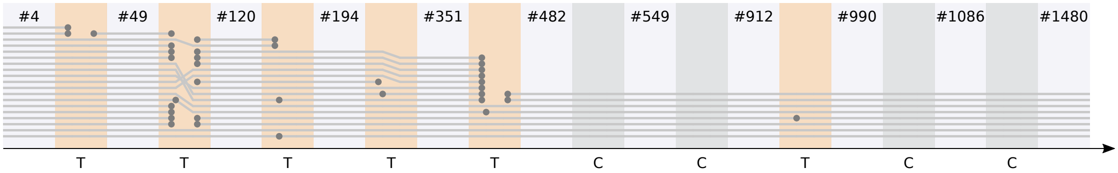
The plots also show noticeable regularities in the way transformations vary. To discuss these we distinguish between the two axes of Fig. 3.10 as two scales of the representation, each of which corresponds to a different set of events. First the horizontal axis: an event at this level is the bulk transformation or conservation of an utterance by a subject, without going into the detail of the way an utterance changes. This yields a series of conservation and transformation events, one at each depth in the branch. Call this the branch dimension, pictured in Fig. 3.11a. A salient feature on this dimension is the apparent burstiness of transformations. Since successive subjects perform transformations independently, confirming this trend would indicate a behaviour reminiscent of punctuated equilibria: a transformation occurring after a period of stability would result in a new utterance that is more likely to be transformed again. We quantify this trend further down.

Second, the vertical axis which delves into the detail of a transformation represented as a set of word insertions, deletions, conservations and replacements with or without exchange. Call this the utterance dimension. An important feature of this representation of transformations is its uniqueness. Indeed, at the mathematical level a consensus alignment encodes a transformation as a pair of word sequences with gaps (and possible sub-alignments of exchanged parts), an encoding that is not unique. Insertions and deletions that happen together can be reordered (putting insertions before their neighbouring deletions instead of the other way around, or alternating an insertion with a deletion); The exchange of two parts around a stable chunk can also be re-encoded by inverting the roles of stable and exchanged chunks, all without changing the transformation represented by the encoding. By compressing the gaps in this encoding, the utterance dimension merges these equivalent versions together and produces a unique diagram representing the transformation. We picture this correspondence between the transformation diagram in the utterance dimension and the compressed form of consensus alignments in Fig. 3.11b.

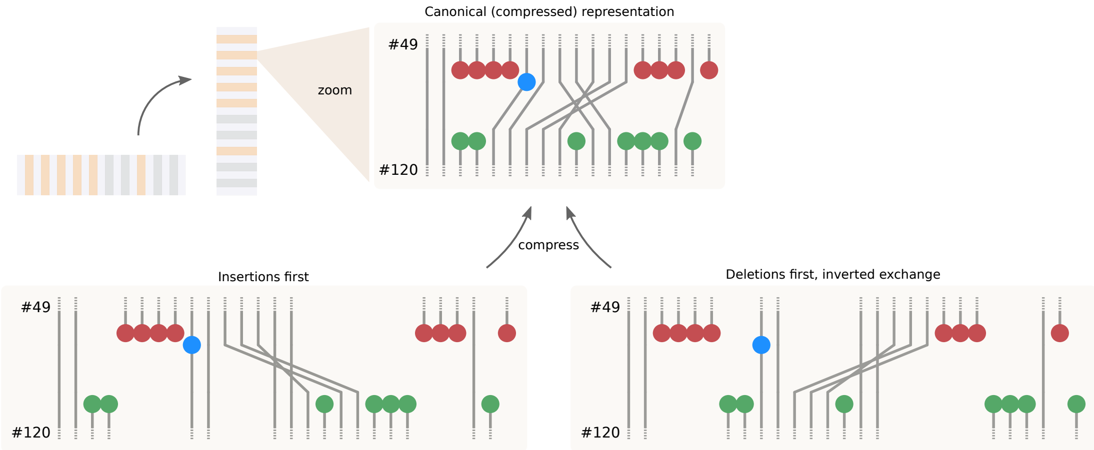
This subtlety in the encoding of transformations is not a coincidence. # TODO: this would be interesting to detail. It relates to the fact that, in spite of the one-dimensional nature of text written on a line, utterance transformation is a multi-level process that can operate on whole groups of words at a time (for instance when insertions and deletions happen together) and does not necessarily reduce to a sequential series of events. Manually inspecting the branches' transformations on the utterance dimension indicated several trends to that effect, several of which are visible in Fig. 3.10:

- Deletions, exchanges, and insertions seem bursty, that is they appear in large chunks (a behaviour that replacements do not seem to have). The bursts also seem longer if the utterance they appear in is longer.
- Insertions seem to rarely occur without deletions. When they appear with deletions, the two tend to be close to each other and of similar magnitude.
- All operations are less frequent at the very beginning of utterances.

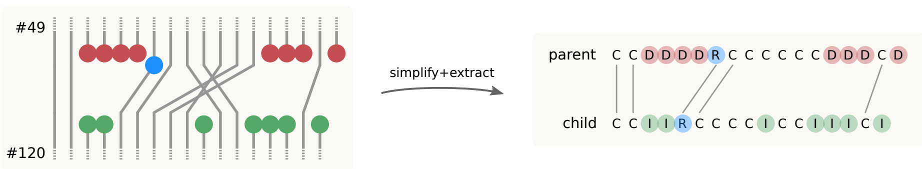
As we just noted, burstiness at the word level is no surprise: words are not processed independently and transforming parts of an utterance is likely to depend on syntactic and semantic boundaries. However, the behaviour of the bursts impose constraints on the kind of model that can account for the transformation process. In particular, to account for the possibility of insertion and deletion bursts that match in size when close to each other, a generative model would need to involve memory and attention span mechanisms that allow bursts to relate to their neighbouring operations (both preceding and following). Similarly, accounting for exchanges with a generative model also requires at least a memory component that is capable of recalling the postponed part of an exchange. Such mechanisms are beyond the score of this chapter, and we therefore focus on devel-



(a) Branch dimension. This level looks at whether or not an utterance is transformed, without going into the detail of changes (hence the greyed out dots and lines). Similar to Fig. 3.10, light grey bands are what subjects see, and the bands between those represent what the subjects do with what they read. An orange band indicates that an utterance was transformed, that is a T event, and a dark grey band indicates that an utterance was perfectly conserved, that is a C event. The corresponding ordered series of events is shown underneath the axis' arrow.



(b) Utterance dimension. This level looks at the detail of a transformation, and represents it with a diagram that compresses the pair of sequences produced by aligning parent and child utterances. This diagram uniquely represents the transformation, and merges any variations in encoding that can exist in pairs of sequences with gaps. The top level of the figure shows how the canonical representation comes from the lineage plots. The bottom level shows two equivalent encodings of the same transformation (as would be produced by the alignment tool), which compress to the same canonical representation.



(c) Parent and child arrays of operations. The canonical representation is further simplified by discarding the change in position encoded by word exchanges, and only keeping the information on whether a word is conserved or replaced. The procedure results in two arrays of word operations: a parent array made of conservations (C), replacements (R) and deletions (D), and a child array made of conservations, replacements and insertions (I). Conservations and replacements in the parent array, if not involved in exchanges, are linked to their corresponding operation in the child array, such that we can compute the distance between a block of insertions in the child and a block of deletions in the parent (except when exchanges separated the blocks).

Figure 3.11: Analysis dimensions. Transformations are analysed along two dimensions. The branch dimension only looks at whether utterances are transformed or not, thus sees a series of T (transformed) and C (conserved) events. The utterance level looks at the detail of the transformations, and after simplification represents them with two arrays of operations, one for the parent utterance (made of C, R and D operations) and one for the child utterance (made of C, R and I operations). This example is built on branch #49 from Fig. 3.10. # TODO: Make this work in grayscale

oping a descriptive—rather than generative—model. While it will not allow for a reconstruction of the transformation process, this approach will provide a synthetic understanding of the transformation behaviour without needing to rely on cognitive mechanisms. By abstracting out the basic building blocks of transformations, we will then be able to gradually increase the level of detail with which we understand the regularities of their interactions.

Descriptive transformation model

Our model relies on a simplification of the transformation diagrams in the utterance dimension of lineage plots, which we take to be the canonical representation of a transformation. In order to keep the model palatable, we first set aside part of the information provided by exchanges. Indeed, the natural way of analysing an exchange in a transformation diagram is to see it as a permutation of a sub-sequence of words in the utterance, with possible replacements, insertions and deletions added in-between. Analysing the regularities of such a process is matter for a model in itself, and we chose to leave this aspect of transformations for further research. We instead focus on insertions, deletions, and replacements, and keep from exchanges only the conserved or replaced status of a word. Note that while this excludes any shifts in position from our model, the approach still benefits from having detected exchanges earlier in the procedure: it guarantees that the remaining insertions and deletions correspond to actual appearances and disappearances, not undetected exchanges.

From a given transformation diagram we then extract two arrays of word-level operations,[20] one for the parent utterance and one for the child utterance. The parent array contains conservation, replacement and deletion operations, and the child array conservation, replacement and insertion operations. The transformation diagram further provides us with the correspondence of conservation and replacement operations between the two arrays (except for operations that were involved in an exchange, for which we lose position information), such that we can measure the distance between two blocks of insertions and deletions (except if the two blocks are separated by operations involved in an exchange).

Fig. 3.11c illustrates this simplification of transformations, which we use as our model for the process: it represents transformation as two arrays of word-level operations, one for the parent utterance made of word conservations, replacements, and deletions, and one for the child utterance made of word conservations, replacements, and insertions. Operations that happen on several contiguous words are called chunks. Conservations and replacements in one array can additionally, but not necessarily, be paired with another conservation or replacement in the other array. When insertion and deletion chunks are separated by paired conservations or replacements, it is then possible to define the distance between the two chunks of operations as the number of conservations or replacements separating the two. When unpaired conservations or replacements separate an insertion and a deletion chunk, this distance is undefined.

3.4.4 Model refinement

Having defined our model for transformations, we now delve into the detailed behaviour that it captures. We do so in three stages. First, we quantify the extent to which transformations are bursty, both in the branch dimension and in the detailed transformation model (utterance dimension). In

[20] We use the phrase "array of operations", and not "series of events", to emphasise that these operations exist on the one-dimensional utterance axis, but do not necessarily come from a sequential generation process. The two terms refer to the same mathematical object, and simply change the interpretation of the index: for a series of events the index represents time, for an array of operations it does not.

doing so we establish the prevalence of operation chunks in the transformation model. We then characterise the number of individual and chunk-level operations that occur in utterances, linking their magnitude and probability to the length of the parent utterance and the position at which they occur. Finally, we examine the dependencies between each operation type, and highlighting a close relationship between insertions and deletions.

Bursty behaviours

We begin by measuring the extent to which each dimension features bursty behaviour. Following Jo et al. (2012; who rely on Goh and Barabási 2008), we measure the burstiness of a series of events through the parameter B defined as

$$B = \frac{\sigma_{intervals} - \mu_{intervals}}{\sigma_{intervals} + \mu_{intervals}}$$

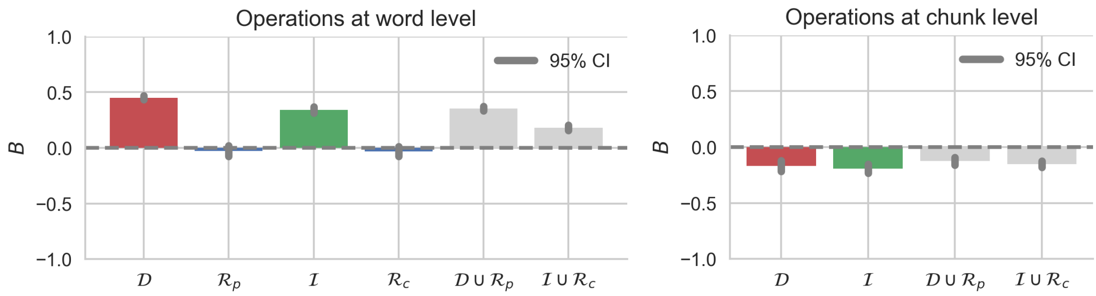
where $\sigma_{intervals}$ and $\mu_{intervals}$ are respectively the standard deviation and mean of the distribution of inter-event times in the series of events. The same computation applies to arrays of operations (the two have the same mathematical description). B has values between -1 and 1; $B = -1$ corresponds to a perfectly regular process ($\sigma_{intervals} = 0$, and $\mu_{intervals} > 0$ is the constant period of events), $B = 0$ indicates a burstiness equivalent to that of a Poisson process, where the occurrence of a new event does not depend on the presence of previous events (and $\sigma_{intervals} = \mu_{intervals}$), and $B = 1$ corresponds to an asymptotically perfectly bursty process (it is the limit ${}^{\mu_{intervals}}/_{\sigma_{intervals}} \to 0$). Intuitively, a process with average inter-event time shorter than its standard deviation will often have events close to each other with a few long intervals without events, and a process with an average inter-event time longer than its standard deviation will have events more evenly spaced relative to their mean spacing.

In the branch dimension, an event is the transformation of an utterance, and the absence of event is the conservation of an utterance. Note that our data in this dimension is truncated due to branches not being infinite. When the last subject in a branch does not transform the utterance they reproduce, we do not observe the actual duration of that stability: had the branch continued, the stability could have been interrupted immediately, or could have lasted for many more reproductions of the utterance. Including these truncated intervals in the distribution of inter-event times artificially inflates the burstiness (because it adds underestimated intervals to the distribution), but removing them biases our sample towards inter-event times for longer utterances (earlier in the branch), which could also inflate burstiness. We thus present measures for both distributions, with and without the truncated intervals.

Burstiness in the branch dimension with truncated intervals is $B_{branch,all} = 0.252 \pm 0.029$, and burstiness without the truncated intervals is $B_{branch,observed} = 0.304 \pm 0.031$ (both error estimates correspond to the 95% confidence interval based on Student's t-distribution, considering each tree as an independent burstiness measure). Both measures show that the transformation process in the branch dimension is significantly bursty. This is consistent with our intuition that when a transformation appears after a period of stability, it is likely to trigger other transformations following it until a new stable (often much shorter) utterance is reached.

The situation in the utterance dimension transformation model involves more event types. In the parent array, we note the series of deletion events $\mathcal{D}$ and the series of replacements $\mathcal{R}_p$. In the child array, we note the series of insertion events $\mathcal{I}$ and the series of replacements $\mathcal{R}_c$. A conserved word is considered an absence of event. Note that because of inserted and deleted words, replacements may not appear with the same distributions in the parent and child arrays. As a consequence, $\mathcal{R}_p$ and $\mathcal{R}_c$ may not have the same distribution of inter-event times. The burstiness measures for each of these series are shown in Fig. 3.12a, along with the burstiness of the series made of all parent or child events without distinguishing their type. The plots show that deletions and insertions are both bursty, while replacements are undistinguishable from a non-bursty process such as a Poisson process. When all event types are joined together, the process is also bursty, albeit slightly less.

Given the strength of this behaviour for deletions and insertions, we further look at these series by collapsing each contiguous chunk of deleted or inserted words into a single event. This leads to a series of deletion and insertion chunks separated by word replacements and word conservations (non-events). For inter-event times, it corresponds to removing the null values in the previous distributions of inter-event times (which separated words in the same chunk); computing the burstiness of the chunk process is therefore straightforward. The values plotted in Fig. 3.12b show that none of the chunk processes are bursty; rather, they are slightly more regular than a Poisson process would be.



(a) Burstiness of word operations

(b) Burstiness of chunks

Figure 3.12: Burstiness of operations in the utterance dimension. The left pane shows the burstiness of each type of word-level operation in parent and child arrays, as well as the burstiness of the series made of all operations joined regardless of their type. The right pane shows the burstiness for deletions, insertions, and joined events, where contiguous blocks of operations are collapsed into single events. This corresponds to the burstiness of *chunks* of deletions, insertions, and joined events (i.e. only considering strictly positive inter-event times). Grey lines are the 95% confidence intervals based on Student's t-distribution, considering each tree as an independent burstiness measure. # TODO: FIXME: is it okay that here we count each tree as an independent measure, whereas in what follows we count each transformation as an independent measure?

Although this behaviour is consistent with our intuition of the way an utterance is reformulated, there is a question as to whether the alignment procedure does not favour burstiness. Indeed, the scores of operations are parametrised in such a way that insertion and deletion gaps are assigned different costs for initial opening and extension. However, while this parametrisation makes it possible for burstiness to be more easily identified, it does not make it a necessity: setting the gap opening cost to the same value as the gap extension cost would make the alignment tool neutral with respect to burstiness (setting it lower would be biased against burstiness, and the alignment algorithm would favour word mismatches over gaps to encode differences). In our case, the parameters we trained set the gap opening cost to a much higher value than the extension cost (.29 vs. .12 in absolute values), such that the alignment tool does find bursty insertions and deletions more easily. However, these parameters are learned from hand-coded alignments and their output has been validated on test samples: any bursty insertions or deletions detected by the alignments is therefore the product of the data itself.

Position and utterance length

The general trends presented in Section 3.4.1 indicated that utterance length has a strong effect on the probability and magnitude of transformations. The transformation models now lets us explore in detail the way word and chunk operations depend on the size of an utterance, on one side, and on the position at which they occur, on the other.

We begin by looking at the probability of each operation as a function of utterance length. Fig. 3.13a plots the logistic regression of the presence or absence of deletions, insertions, and replacements as a function of the number of words in the parent utterance. The length of the parent utterance has a significant effect on all three operations, with deletions being the quickest to increase in probability, followed by replacements then insertions: the threshold for having deletions over half the time is 19 words, 22.6 for replacements, and 28.1 words for insertions; the slopes of the regressions are also ordered this way. In other words, a longer utterance will have a higher risk for all operations, and the increase is strongest for deletions, then for replacements, then for insertions.

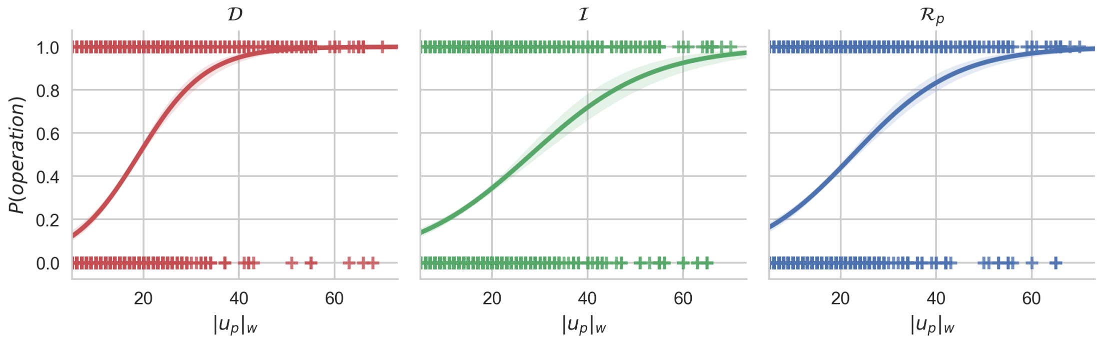
Fig. 3.13b further shows the number of operations as a function of parent utterance length, either counting one for each word affected or counting one for each contiguous chunk of words affected. The number of word and chunk operations increase close to linearly as a function of parent length. Deletions have by far the strongest link to parent length, both at the word and chunk levels, followed by insertions then replacements. Note that the replacement counts barely change between word and chunk level since this operation is not bursty: it affects mostly isolated words instead of chunks of words. In short, a longer utterance has a higher probability of suffering any type of operation, with on average over a quarter of the words deleted, the equivalent of a fourteenth of the original utterance in new words, and about a twentieth of the words replaced.

Manual exploration of the lineage plots also indicated that operations are not positioned evenly in the utterances. To quantify this behaviour we apply the susceptibility measure developed in the previous chapter to positions in an utterance. For words at position $x \in [0, 1]$ relative to their utterance's length ($x = 0$ for words at the beginning, $x = 1$ for words at the end), the susceptibility $\sigma_O(x)$ to an operation $O \in \{D, I, R\}$ is defined as the ratio of $s_O(x)$, the number of times words at relative position x are the target of operation O, to $s^0_O(x)$, the number of times those words would be the target of operation O if the choice of words were random:[21]

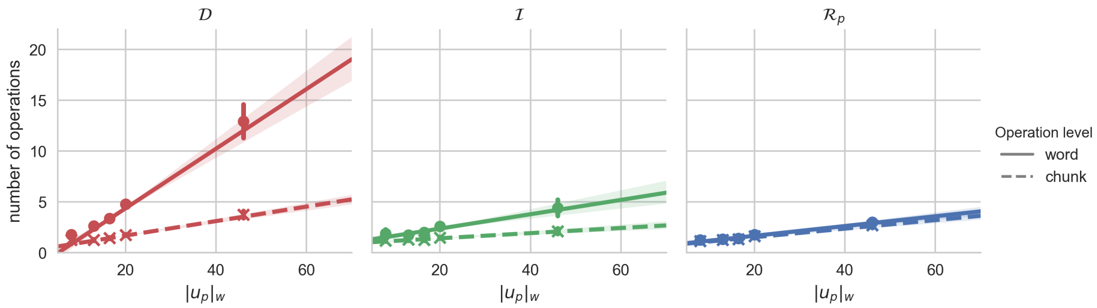
$$\sigma_O(x) = \frac{s_O(x)}{s^0_O(x)}$$

Fig. 3.14 plots σ_D, σ_I and σ_R (for replacements on the parent side) both overall and for binned parent lengths. The leftmost plots show that deletions and insertions are half as likely to appear at the very beginning of an utterance as they would at random, and more likely than random in the second half of an utterance. This is consistent with the well-known primacy effect in recall of word lists. In this case, subjects transform the beginning of an utterance on average much less than the rest. Replacements feature this primacy effect to a lesser extent, with the addition of a slight recency effect: words at the end of an utterance are slightly less replaced than non-extremity words. The

[21] Since operations in a given transformation are not independent, we scale both $s_O(x)$ and $s^0_O(x)$ such that each transformation has a maximum contribution of 1 to the total counts. This procedure is similar to the susceptibility scaling approach we followed in the previous chapter.



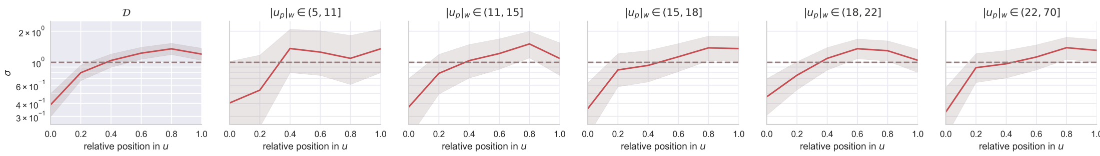
(a) Probability of word operations w.r.t. parent length, computed as the log-odds logistic regression of the presence or absence of a given operation in the transformation of u_p (parent) into u_c (child), versus the number of words in u_p. Colours correspond to the colour-coding used in Fig. 3.10. Light shades are 95% regression confidence intervals.



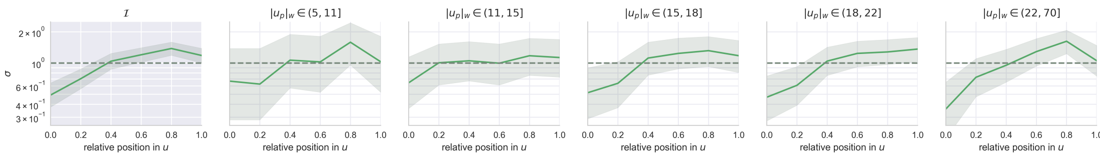
(b) Number of word and chunk operations w.r.t. parent length. Parent lengths are binned into 5 quantile-based bins. Word-level counts the number of individual words affected by an operation (deletion, insertion, replacement). Chunk-level counts the number of contiguous chunks of words affected by an operation. Light shades are 95% regression confidence intervals, and vertical bars are 95% confidence intervals for the value of a bin (Student t-based, here counting each operation as an independent measure # TODO: FXME: vertical bars should count each transformation as independent).

Figure 3.13: Probability and number of word-level or chunk-level operations. # TODO: add raw distributions?

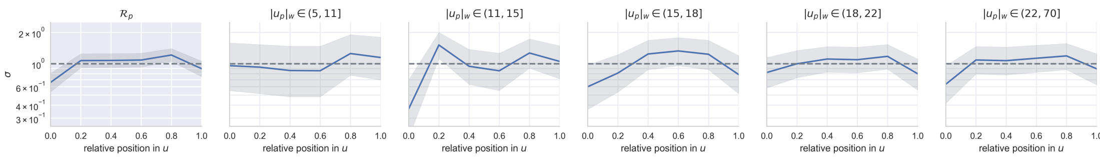
plots at binned parent lengths show little to no variation in these patterns: each pattern is more or less marked depending on the parent sentence length (especially for replacements, which seem more uniform for short utterances), but the general behaviour is the same for different parent lengths.



(a) Deletions



(b) Insertions



(c) Replacements

Figure 3.14: Susceptibility for word operations as a function of relative position in the utterance. The leftmost plot of each sub-figure (blue background) shows σ computed over all transformations. The plots with the white backgrounds show σ computed over transformations with binned parent utterance lengths, indicated in the plot titles. Parent length bins are quantile-based, that is computed to have the same number of utterances in each bin (the bins are identical to Fig. 3.13b). Light shades are the 95% confidence intervals computed following the Goodman (1965) method for multinomial proportions, considering each transformation as an independent measure.

Finally, we examine the dependence of operation chunk size on its position in an utterance. The manual exploration of lineage plots did not hint to any effect at this level, but the question now appears legitimate: since subjects delete words on average more often towards the end of an utterance, it is possible that those deletions are also longer if they correspond to larger memory loss. Fig. 3.15 shows the dependence of chunk size on position in the utterance, for deletions, insertions and replacements, both overall and for binned parent length. Deletions exhibit a slight effect of position on chunk size, which is significant for parent lengths between 11 and 15 words.[22] That is, for those lengths, deletions towards the end of the utterance are significantly larger than deletions at the beginning (4.1 words versus 1.7 words on average), in addition to being more frequent (see the susceptibility plots above). The trend is present for deletions at all lengths, though most of the time not significative. Other operations do not seem to exhibit this behaviour (the variations for insertions are not significative).

Overall, these measures show that deletions are more frequent than insertions, which are more fre-

[22] The plots also indicate that the overall chunk size increases with parent length, a slight effect which was confirmed for deletions and insertions with dedicated regressions, but which we do not discuss further given its mildness (slopes respectively .030 and .013, both significative with $p < .001$).

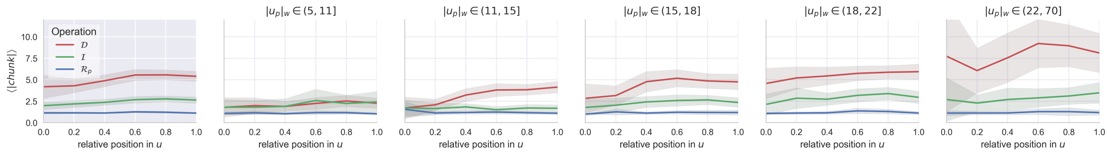


Figure 3.15: Chunk size w.r.t. parent length and position in utterance. The leftmost plot (blue background) shows the average chunk size w.r.t. parent length for all utterances. The plots on its right (white background) divide that data into binned parent lengths (bins identical to Figs. 3.13b and 3.14). In each plot, the height of a line for a given relative position x corresponds to the average size of the chunks in which words at position x are found; for instance, a deletion chunk that spans the second half of an utterance will be spread over $x \in [.5, 1]$. Average sizes are weighted such that each utterance contributes 1 unit. Light shades are the 95% confidence intervals (Student t-based, considering each transformation as an independent measure).

quent than replacements. Operations happen preferentially in the second half of utterances (except replacements which favour all positions except extremities), and their number of words and number of chunks are proportional to the parent length. Deletion chunks are also larger in the second half of utterances, compared to in the first half.

Dependencies between operations

Manual exploration of the lineage plots indicated that operations have non-trivial dependencies between each other. The contingency table combining the presence or absence of each operation gives an overview of these dependencies:

		Deletion			
		no		*yes*	
Replacement		*no*	*yes*	*no*	*yes*
Insertion	*no*	1381	415	286	308
	yes	66	94	399	512

Fig. 3.16 illustrates this data with a mosaic plot, rendering some of the trends more visible. One way to look at these figures is by considering deletions first. Without deletions, insertions are very unlikely (8.2%), and replacements are also unlikely (though less so: 26.0%): the most likely event without deletion is by far a transformation with no change at all (70.6%). With deletions, all four possibilities are of comparable probabilities: having both insertions and replacements is the most likely case (34.0%), followed by insertions without replacements (26.5%), then replacements without insertions (20.5%), then neither replacements nor insertions (19.0%). Overall, deletions can be seen as a gate for other transformations: without them the most likely outcome is no change at all, with them all situations have relatively similar probabilities. A second way to look at the contingencies is to consider that insertions trigger deletions: without insertions, deletions happen only 24.9% of the time, whereas with them deletions are extremely likely (85.1%). Replacements are also linked to insertions, either with or without deletions: the presence of one always increases the probability of the other.

The process is joint of course, and separating it into different stages would require more knowledge of the cognitive mechanisms that underlie these transformations. In spite of this, the relationship between insertions and deletions seems to be well constrained, a fact we see not only in the probability of presence or absence, but also in the number of operations inside a given transformation. The link between insertions and deletions can be seen by plotting the distribution of the number of insertions conditioned on the presence of deletions, and vice-versa. Both plots are shown on Fig. 3.17: aside from being less probable, insertions without deletions are also much smaller in number compared to with deletions. A similar behaviour is observed in the opposite case: deletions that happen in the presence of an insertion are much greater in number than without insertions.

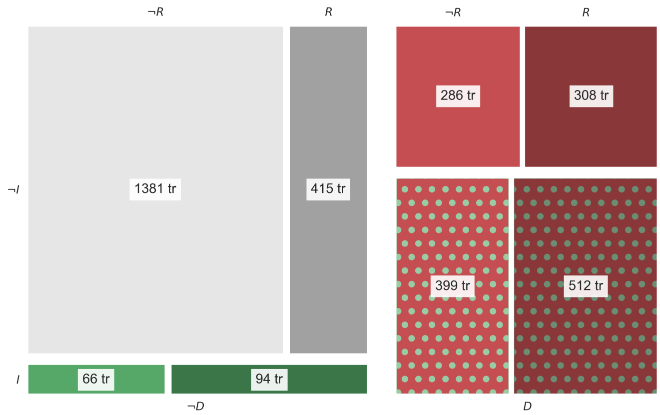
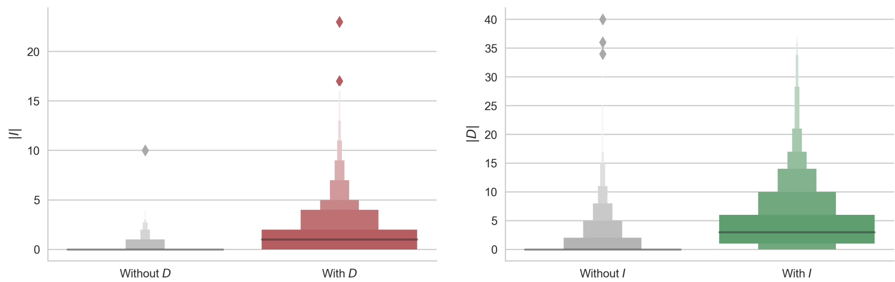


Figure 3.16: Mosaic plot of the contingency table between deletions, insertions, and replacements. Red rectangles indicate deletions are present; green rectangles or green dots indicate insertions are present; darker colours indicate replacements are present. Each rectangle also indicates the number of transformations it represents (corresponding to the rectangle area).



(a) Number of insertions conditioned on the presence of deletions. (b) Number of deletions conditioned on the presence of insertions.

Figure 3.17: Letter-value plots (Hofmann, Kafadar, and Wickham 2011) of deletion and insertion counts conditioned on the presence of one another. In a given plot, the boundaries between boxes are placed at the $^1/_{2^i}$-th quantiles: the middle line is the median, and above and below it the biggest box stops at the first and third quartiles, the second biggest stops at the first and seventh 8-quantiles (octiles), and so on and so forth. Diamonds are outliers that do not fit into the smallest box.

Deletions and insertions thus seem closely linked, as our intuition of the process suggests: deletions could be the first manifestation of the subject having forgotten something in the parent utterance, and their presence then opens the door to further reformulations, possibly to make up for the forgotten content.

This relates to the last observation produced by our manual exploration: insertions and deletions seem to occur in similar sizes when close to one another. To quantify this observation we estimate a correlation function between the sizes of insertion and deletion chunks separated by fixed numbers of conserved or replaced words. More precisely, for each insertion chunk in the data set we identify the nearest deletion chunk either before or after it, separated by words that are not involved in an exchange.[23] If an insertion chunk has such a nearest neighbour (it may not if there were no deletions, or if it occurred in the middle of exchanged words such as in Fig. 3.11c), we note r the separation between the two chunks. If insertion and deletion chunks face each other, $r = 0$; otherwise, $r < 0$ if the deletion comes before the insertion in the utterances, $r > 0$ if the deletion comes after, and $|r|$ equals the number of conserved or replaced words separating the two. For a given value of r, we compute a robust linear regression of insertion chunk size against deletion chunk size for all insertion-deletion chunks separated by r. We then take the slope of that regression as an indicator of the correspondence between the sizes of r-separated insertion and deletion chunks.[24]

Fig. 3.18 shows the robust regressions and the estimated correlation function for $r \in \{-5, ..., 5\}$ (outside of which there was always less than 10 insertion-deletion couples). The plot shows three important points. First, the vast majority of nearest neighbours insertion and deletion chunks face each other ($r = 0$), and their sizes significantly correlate. Second, the correlation initially decreases to become non-signifcant as $|r|$ increases. The third and most interesting point is that the correlation function is skewed towards the left: it is significantly above zero for $r = -1$ but not for $r = 1$, then also for $r = -4, -5$ at higher values than for $r = 4$. Note however that the last three points represent only 10 to 12 insertion-deletion couples each and are thus more susceptible to outliers (especially to deletion outliers, i.e. the x axis, which the M-estimation technique we used does not counter). Overall, the correlation is positive for insertion and deletion chunks facing each other, and also often for deletion chunks preceding insertions by a few words.[25] This trend is consistent with the intuition we outlined above, according to which insertion chunks could come as tentative replacements for the content that was lost in the deletions that directly precede them.

The transformation model we introduced thus captures several important behaviours in the way subjects change utterances. Looking at transformations as made of word-level replacements, deletions and insertions, we see that both insertions and deletions are bursty, and that the presence and magnitude of an operation depends strongly on utterance size and the position at which it appears in the utterance. We further see that insertion and deletion chunks are closely related: insertions behave as if they were gated by the presence of a deletion, and their size tend to correlate to that of deletions appearing at the same time or shortly before them.

3.4.5 Lexical feature makeup

TODO: There is space for improvement in this last discussion

[23] As alluded to when introducing the transformation model, when exchanges separate an insertion chunk from a deletion chunk there are several paths from one to the other, depending on when one traverses the exchange; different paths can have different final distances, none of which are more or less plausible than the others. The distance between an insertion chunk and a deletion chunk separated by an exchange is thus not clearly defined.

[24] The robust regression lets us minimise the impact of outliers in the distributions of insertion chunk sizes, which otherwise had a strong effect on more common correlation measures. The regression is computed using the Statsmodels statistics library for Python, which implements robust M-estimation using Huber's T norm (Huber 1981) with a default parameter of 1.345.

[25] The detail of these plots is sensitive to cropping in the data, and especially to constraints on the maximum deletion size since it can remove x-axis outliers. The general trend we observe is always conserved however: deletions preceding insertions correlate more than deletions following insertions.

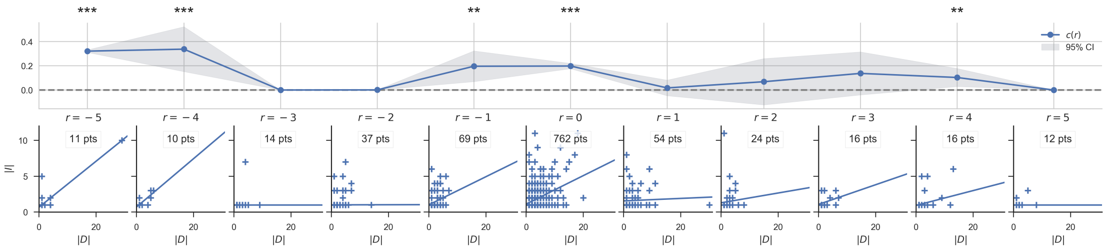


Figure 3.18: Size correlation between nearest-neighbour insertion and deletion chunks at different distances. The bottom subplots show the robust regressions for couples of insertion-deletion chunks separated by a given value of r. The text at the top of each subplot indicates the number of insertion-deletion couples that the subplot represents. The top plot shows the values of the regression slopes aligned to the bottom subplots, with 95% regression confidence intervals and star-coded significance levels (*** for $p < .001$, ** for $p < .01$, * for $p < .05$ and nothing otherwise).

We finally descend to the lower level of lexical word features to characterise the words involved in insertions, deletions and replacements. To do so we extend the feature analysis developed in the previous chapter to our current situation. We begin with word susceptibilities, then continue with feature variation upon replacement, and finally show how the accumulation of transformations along the branches leads the lexical makeup of utterances to gradually evolve.

Word features

The situation is parallel to that of the previous chapter, and its analysis can be directly applied. We measure the susceptibility of words to being the target of an operation (either by deletion or replacement) and to being the new word of an operation (either as replacing word or inserted word) in a similar manner to substitution susceptibility. For a given grouping of words g (e.g. grammatical category or feature value), we compute its susceptibility σ_g^- to being a target and its susceptibility σ_g^+ to newly appearing as the ratio of the number of times it is a target (s_g^-) or a new word (s_g^+) to the number of times it would be if the process were a random sampling from the available utterances (s_g^0):

$$\sigma_g^- = \frac{s_g^-}{s_g^0} \quad \text{and} \quad \sigma_g^+ = \frac{s_g^+}{s_g^0}$$

In order to render the results more comparable to the previous chapter, in this section we also filter out stopwords in all the utterances. Fig. 3.19 shows POS susceptibilities for being the target or the new word of an operation. The two measures are very close to each other, and similarly to the online case there is little to no effect of the main categories on susceptibility: adjectives are involved at random, nouns appear slightly less than at random, and verbs slightly more. Verbs, nouns and proper nouns are all irrelevant for targeting, but adverbs are targeted very slightly above random. The other categories (adpositions, numerals and particles) total negligible amounts because they are affected by the stopword filter. Overall, the behaviour for targeting is consistent with what we observed in blogspace (the only difference being the trend for adverbs, which are also less present overall in this data set), and the behaviour for appearances indicates a slight bias in favour of verbs and against nouns (appearance susceptibilities were not analysed in the blogspace data set).

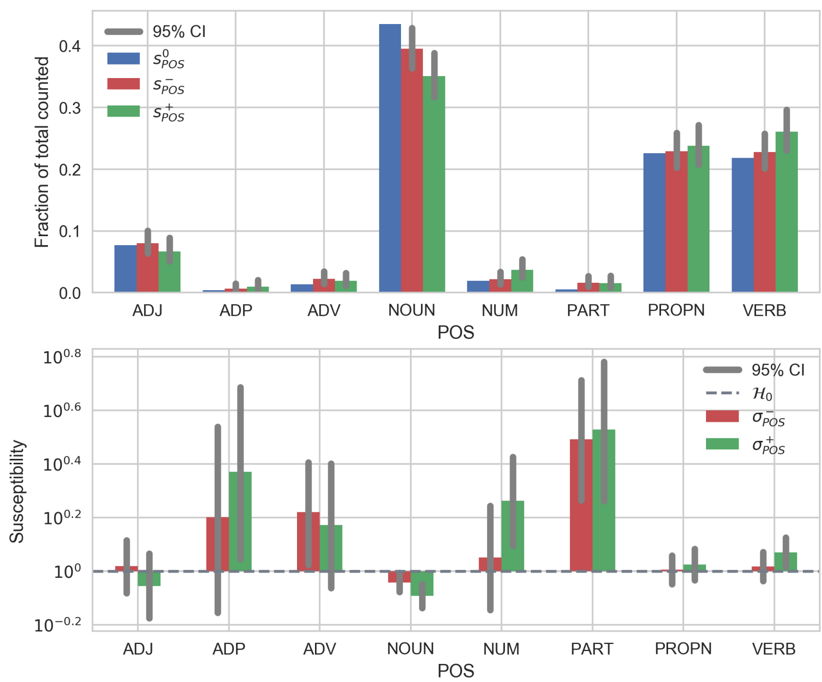
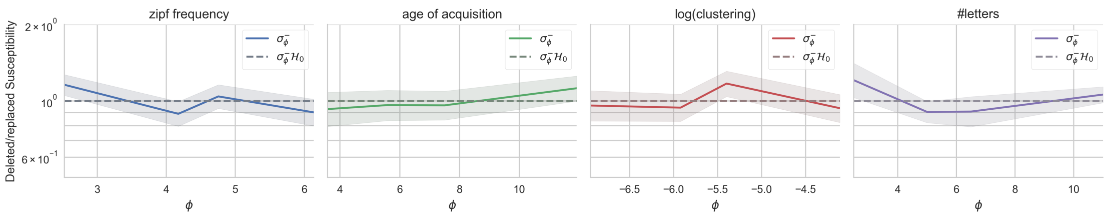
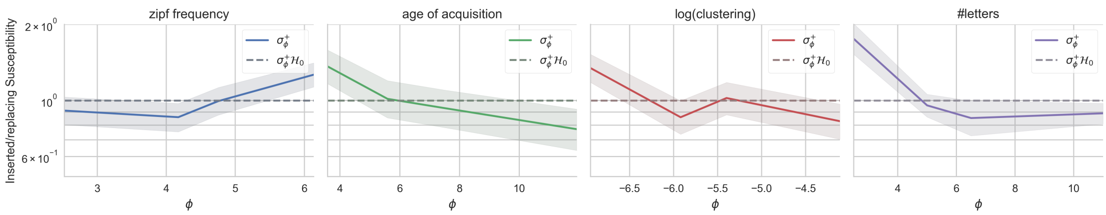


Figure 3.19: POS susceptibility to being replaced or deleted, and to replacing or being inserted. The top panel shows the proportions of POS categories observed in utterances overall (s^0_{POS}), in replaced and deleted words in parent utterances (s^-_{POS}) and in replacing and inserted words in child utterances (s^+_{POS}). The bottom panel shows susceptibilities, that is the ratio of s^-_{POS} and s^+_{POS} to s^0_{POS}. 95% asymptotic confidence intervals are shown in grey (Goodman-based multinomial proportions, considering each transformation as an independent measure). POS tags are from the Universal Dependencies tag set.

For the sake of conciseness, we now focus on the four lexical features that showed relevant effects in the previous chapter: word frequency, age of acquisition, Free Association clustering and number of letters, thus leaving aside number of synonyms and orthographic neighbourhood density. Age of acquisition and clustering are identical to the previous chapter; word frequency was previously computed from the data set itself, and in the present case the overall data set is much smaller. Instead, we relied on external word frequency ratings based on subtitles (Heuven et al. 2014), a source which has repeatedly beaten previous predictors of standard lexical decision times (see Heuven et al. 2014 for more details). These frequencies are provided on what the authors introduce as the Zipf scale, computed as $\log_{10}(\text{Frequency per billion words})$. The frequency values thus use a different source than those of the previous chapter, but their final computation only differs by an affine transformation. Fig. 3.20a plots the feature susceptibilities to targeting and appearance. The trends for frequency, age of acquisition and clustering are consistent with previous results. Low frequency, high age of acquisition words tend to be very slightly more targeted, and clustering is mostly not relevant to the process. Number of letters has a different behaviour than previously, as short words are slightly more targeted than random, beyond the effect for long words. It is unclear where this change of effect comes from, as it could be due to factors ranging from analysis tweaks (e.g. an update to the stopword list) to the fact that subjects could be more inclined to replace some words because of a different task context or a different set of utterances. All these trends are extremely subtle however (much more than in the blogspace data set), and we do not attempt to explain them any further at this point. Fig. 3.20b shows the corresponding feature susceptibilities for appearance, where the trends for frequency and age of acquisition are reversed: more frequent, lower age of acquisition words are more susceptible to appearance. Low clustering and short words appear also more than random, all of which are consistent with the variation patterns observed previously, and which we confirm below.



(a) Susceptibility to targeting.



(b) Susceptibility to appearance.

Figure 3.20: Feature susceptibilities of words to targeting (deletion and replacement on the parent side) and appearance (insertion and replacement on the child side), binned by quartiles, with 95% asymptotic confidence intervals (Goodman-based multinomial, considering each transformation as an independent measure).

Indeed, the analysis of feature variation can also be directly applied to word replacements (though not to deletions or insertions), and Fig. 3.21 shows the results for the current data set. The plots for frequency, age of acquisition and clustering are strikingly similar to previous results. Here too however, number of letters has a different behaviour than previously: instead of a uniform negative bias, ν_ϕ and ν_ϕ^{00} are substantially changed: both are much closer to word conservation ($y = x$) than previously, and their intersections with ν_ϕ^0 and between each other are also closer to each other. In other words, the number of letters of words are better conserved in this data set than in blogspace. Two factors could have influenced this change of effect: first, the alignment procedure favours replacements for closely related synonyms (evaluated by their vector similarity), which could explain the fact that ν_ϕ and ν_ϕ^{00} are much closer to each other and to $y = x$. Second, the fact that ν_ϕ^{00} changes so much from the its values in the previous chapter indicates that the sampling of source utterances also has a role. Recall that in this case ν_ϕ^{00} is the average number of letters of synonyms of words that are replaced: ν_ϕ^{00} being closer to $y = x$ then indicates that synonyms of words in the current utterances are closer in size to their originals than is the case in the blogspace utterances, a fact that could contribute to the overall better conservation of number of letters.

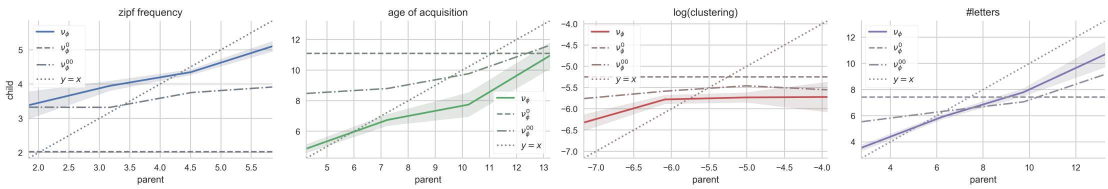
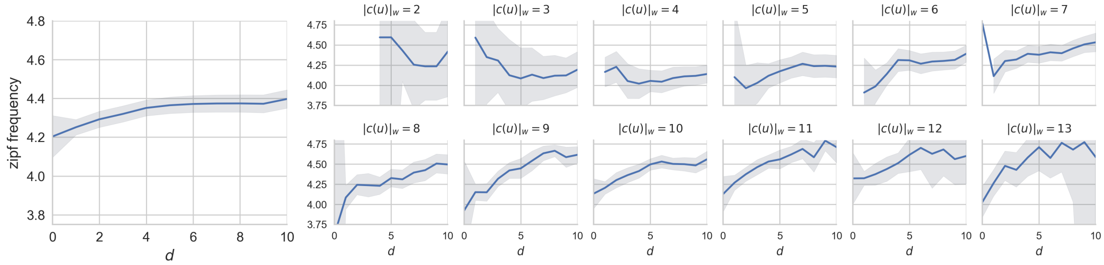


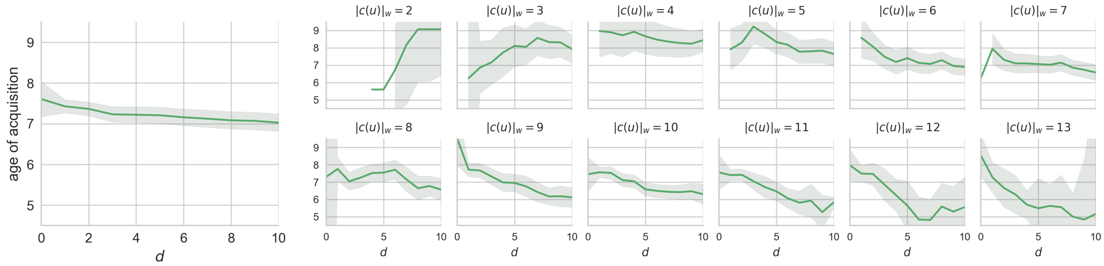
Figure 3.21: Feature variation upon replacement. ν_ϕ, average feature word of the appearing word as a function of the feature value of the targeted word (fixed bins), with 95% asymptotic confidence intervals based on Student's t-distribution. Refer to **??** for the detailed interpretation of the curves.

Branch evolution

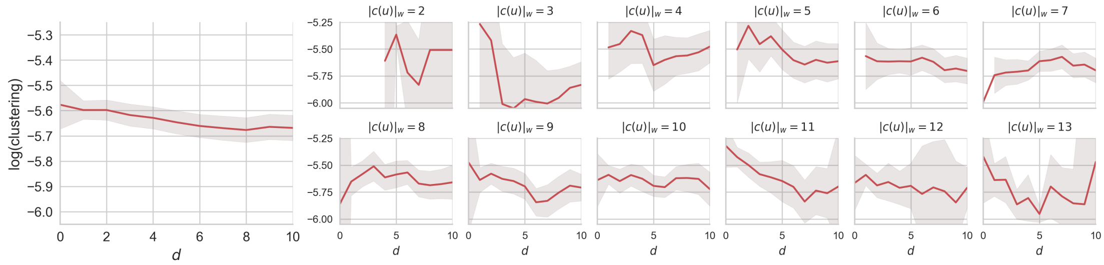
Since these features vary consistently, on average, with each transformation of the utterances, we finally ask if any long-term evolution due to the transformations is observable. We therefore plot the evolution of the average features of utterances as a function of branch depth for word frequency, age of acquisition, clustering and number of letters. Fig. 3.22 plots this information both for all utterances and divided into fixed content lengths. The evolution of each feature is consistent with its susceptibility to targeting and appearance, and its variation upon replacement. Average word frequency significantly increases with depth, both globally and at fixed content length. This fits with low frequency words being more susceptible to targeting and high frequency words more susceptible to appearing (Fig. 3.20), as well as with frequency increasing upon replacement (Fig. 3.21). The reverse is true for age of acquisition, which decreases with depth (albeit significantly for certain content lengths only). Clustering and number of letters both decrease also, though clustering shows no clear trend at fixed content lengths and its evolution might therefore be due to utterance shortening. It is worth noting that for number of letters, in spite of a small targeting bias in favour of short words, the much stronger appearance bias in favour of short words wins in the long run: average number of letters decreases along the branches, even at fixed content length. While the trends are not strong, it is still noteworthy that they are visible at the level of utterance averages: in less than 10 iterations, transformations which mostly maintain the overall meaning of the utterances have a significant effect on these features, beyond the shortening of utterances (and consequent removal of words that could have an effect on the features). Through transformations, subjects thus gradually evolve the utterances to use more frequent, shorter words, learned earlier and with lower free association clustering coefficients.



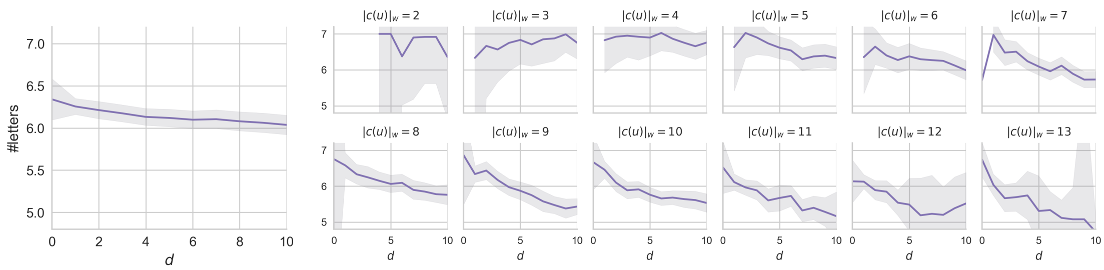
(a) Word frequency



(b) Age of acquisition



(c) Clustering



(d) Number of letters

Figure 3.22: Evolution of average utterance features as a function of depth in the branch, with 95% confidence intervals based on Student's t-distribution (considering each utterance as an independent measure).

3.5 Discussion

TBD

Chapter 4

Discussion

Chapter 5

Conclusion

References

Acerbi, Alberto, and Jamshid J. Tehrani. 2017. 'Did Einstein Really Say That? Testing Content Versus Context in the Cultural Selection of Quotations'. *SocArXiv Preprints*, April. https://osf.io/preprints/socarxiv/x8db2.

Bangerter, Adrian. 2000. 'Transformation Between Scientific and Social Representations of Conception: The Method of Serial Reproduction'. *British Journal of Social Psychology* 39 (4): 521–35. doi:10.1348/014466600164615.

Barrett, Justin L., and Melanie A. Nyhof. 2001. 'Spreading Non-Natural Concepts: The Role of Intuitive Conceptual Structures in Memory and Transmission of Cultural Materials'. *Journal of Cognition and Culture* 1 (1): 69–100. doi:10.1163/156853701300063589.

Bartlett, Sir Frederic Charles. 1995. *Remembering: A Study in Experimental and Social Psychology*. Cambridge University Press.

Bastian, Mikaël, Sébastien Lerique, Vincent Adam, Michael S. Franklin, Jonathan W. Schooler, and Jérôme Sackur. 2017. 'Language Facilitates Introspection: Verbal Mind-Wandering Has Privileged Access to Consciousness'. *Consciousness and Cognition* 49 (March): 86–97. doi:10.1016/j.concog.2017.01.002.

Bebbington, Keely, Colin MacLeod, T. Mark Ellison, and Nicolas Fay. 2017. 'The Sky Is Falling: Evidence of a Negativity Bias in the Social Transmission of Information'. *Evolution and Human Behavior* 38 (1): 92–101. doi:10.1016/j.evolhumbehav.2016.07.004.

Boyd, Robert, and Peter J Richerson. 1985. *Culture and the Evolutionary Process*. Chicago: University of Chicago Press.

Brysbaert, Marc, Michaël Stevens, Paweł Mandera, and Emmanuel Keuleers. 2016. 'How Many Words Do We Know? Practical Estimates of Vocabulary Size Dependent on Word Definition, the Degree of Language Input and the Participant's Age'. *Frontiers in Psychology* 7 (July). doi:10.3389/fpsyg.2016.01116.

Cavalli-Sforza, L. L, and Marcus W Feldman. 1981. *Cultural Transmission and Evolution: A Quantitative Approach*. Princeton, N.J.: Princeton University Press.

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

Claidière, Nicolas, Thomas C. Scott-Phillips, and Dan Sperber. 2014. 'How Darwinian Is Cultural Evolution?' *Philosophical Transactions of the Royal Society B: Biological Sciences* 369 (1642): 20130368.

doi:10.1098/rstb.2013.0368.

Claidière, Nicolas, Kenny Smith, Simon Kirby, and Joël Fagot. 2014. 'Cultural Evolution of Systematically Structured Behaviour in a Non-Human Primate'. *Proceedings of the Royal Society of London B: Biological Sciences* 281 (1797): 20141541. doi:10.1098/rspb.2014.1541.

Cock, Peter J. A., Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, et al. 2009. 'Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics'. *Bioinformatics* 25 (11): 1422–3. doi:10.1093/bioinformatics/btp163.

Cornish, Hannah, Kenny Smith, and Simon Kirby. 2013. 'Systems from Sequences: An Iterated Learning Account of the Emergence of Systematic Structure in a Non-Linguistic Task'. In *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, edited by Markus Knauff, Michael Pauen, Natalie Sebanz, and Ipke Wachsmuth. Austin, TX: Cognitive Science Society.

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

Danescu-Niculescu-Mizil, Cristian, Justin Cheng, Jon Kleinberg, and Lillian Lee. 2012. 'You Had Me at Hello: How Phrasing Affects Memorability'. In *ACL '12 Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers*, edited by Haizhou Li, Chin-Yew Lin, and Miles Osborne, 1:892–901. Stroudsburg, PA: ACM. http://arxiv.org/abs/1203.6360.

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

Eriksson, Kimmo, and Julie C. Coultas. 2014. 'Corpses, Maggots, Poodles and Rats: Emotional Selection Operating in Three Phases of Cultural Transmission of Urban Legends'. *Journal of Cognition and Culture* 14 (1): 1–26. doi:10.1163/15685373-12342107.

Fénéon, Félix, and Luc Sante. 2007. *Novels in Three Lines*. New York, NY, USA: New York Review Books.

Goh, K.-I., and A.-L. Barabási. 2008. 'Burstiness and Memory in Complex Systems'. *EPL (Europhysics Letters)* 81 (4): 48002. doi:10.1209/0295-5075/81/48002.

Goodman, Leo A. 1965. 'On Simultaneous Confidence Intervals for Multinomial Proportions'. *Technometrics* 7 (2): 247–54. doi:10.1080/00401706.1965.10490252.

Heath, Chip, Chris Bell, and Emily Sternberg. 2001. 'Emotional Selection in Memes: The Case of Urban Legends'. *Journal of Personality and Social Psychology* 81 (6): 1028–41. doi:10.1037/0022-3514.81.6.1028.

Heuven, Walter J. B. van, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. 'SUBTLEX-UK: A New and Improved Word Frequency Database for British English'. *The Quarterly Journal of Experimental Psychology* 67 (6): 1176–90. doi:10.1080/17470218.2013.850521.

Hofmann, Heike, Karen Kafadar, and Hadley Wickham. 2011. 'Letter-Value Plots: Boxplots for Large Data'. had.co.nz.

Huber, Peter J. 1981. *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. New York, NY, USA: Wiley. http://public.eblib.com/choice/publicfullrecord.aspx?p=224924.

Jo, Hang-Hyun, Márton Karsai, János Kertész, and Kimmo Kaski. 2012. 'Circadian Pattern and Burstiness in Mobile Phone Communication'. *New Journal of Physics* 14 (1): 013055. doi:10.1088/1367-

2630/14/1/013055.

Kashima, Yoshihisa. 2000. 'Maintaining Cultural Stereotypes in the Serial Reproduction of Narratives'. *Personality and Social Psychology Bulletin* 26 (5): 594–604. doi:10.1177/0146167200267007.

Keuleers, Emmanuel, and Marc Brysbaert. 2010. 'Wuggy: A Multilingual Pseudoword Generator'. *Behavior Research Methods* 42 (3): 627–33. doi:10.3758/BRM.42.3.627.

Keuleers, Emmanuel, Michaël Stevens, Paweł Mandera, and Marc Brysbaert. 2015. 'Word Knowledge in the Crowd: Measuring Vocabulary Size and Word Prevalence in a Massive Online Experiment'. *The Quarterly Journal of Experimental Psychology* 68 (8): 1665–92. doi:10.1080/17470218.2015.1022560.

Killingsworth, Matthew A., and Daniel T. Gilbert. 2010. 'A Wandering Mind Is an Unhappy Mind'. *Science* 330 (6006): 932–32. doi:10.1126/science.1192439.

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

Lauf, Aurelien, Mathieu Valette, and Leila Khouas. 2013. 'Analyzing Variation Patterns in Quotes over Time'. *Research in Computing Science* 70: 223–32. http://www.micai.org/rcs/2013_70/Analyzing%20Variation%20Patterns%20In%20Quotes%20Over%20Time.html.

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.

Lombardi, Linda, and Mary C Potter. 1992. 'The Regeneration of Syntax in Short Term Memory'. *Journal of Memory and Language* 31 (6): 713–33. doi:10.1016/0749-596X(92)90036-W.

Mackerron, G., and S. Mourato. 2013. 'Happiness Is Greater in Natural Environments'. *Global Environmental Change*. http://www.sciencedirect.com/science/article/pii/S0959378013000575.

Maxwell, R. S. 1936. 'Remembering in Different Social Groups'. *British Journal of Psychology. General Section* 27 (1): 30–40. doi:10.1111/j.2044-8295.1936.tb00814.x.

Mesoudi, Alex, and Andrew Whiten. 2004. 'The Hierarchical Transformation of Event Knowledge in Human Cultural Transmission'. *Journal of Cognition and Culture* 4 (1): 1–24. doi:10.1163/156853704323074732.

———. 2008. 'The Multiple Roles of Cultural Transmission Experiments in Understanding Human Cultural Evolution'. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 363 (1509): 3489–3501. doi:10.1098/rstb.2008.0129.

Mesoudi, Alex, Andrew Whiten, and Robin Dunbar. 2006. 'A Bias for Social Information in Human Cultural Transmission'. *British Journal of Psychology* 97 (3): 405–23. doi:10.1348/000712605X85871.

Miller, Geoffrey. 2012. 'The Smartphone Psychology Manifesto'. *Perspectives on Psychological Science* 7 (3): 221–37. doi:10.1177/1745691612441215.

Moussaïd, Mehdi, Henry Brighton, and Wolfgang Gaissmaier. 2015. 'The Amplification of Risk in Experimental Diffusion Chains'. *Proceedings of the National Academy of Sciences* 112 (18): 5631–6. doi:10.1073/pnas.1421883112.

Needleman, Saul B., and Christian D. Wunsch. 1970. 'A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins'. *Journal of Molecular Biology* 48 (3): 443–53.

doi:10.1016/0022-2836(70)90057-4.

Niculae, Vlad, and Cristian Danescu-Niculescu-Mizil. 2016. 'Conversational Markers of Constructive Discussions'. *arXiv:1604.07407 [Physics, Stat]*, April. http://arxiv.org/abs/1604.07407.

Norenzayan, Ara, Scott Atran, Jason Faulkner, and Mark Schaller. 2006. 'Memory and Mystery: The Cultural Selection of Minimally Counterintuitive Narratives'. *Cognitive Science* 30 (3): 531–53. doi:10.1207/s15516709cog0000_68.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. 'GloVe: Global Vectors for Word Representation'. In, 1532–43. http://www.aclweb.org/anthology/D14-1162.

Potter, Mary C, and Linda Lombardi. 1990. 'Regeneration in the Short-Term Recall of Sentences'. *Journal of Memory and Language* 29 (6): 633–54. doi:10.1016/0749-596X(90)90042-X.

Rayner, Keith, Timothy J. Slattery, and Nathalie N. Bélanger. 2010. 'Eye Movements, the Perceptual Span, and Reading Speed'. *Psychonomic Bulletin & Review* 17 (6): 834–39. doi:10.3758/PBR.17.6.834.

Zaromb, Franklin M., Marc W. Howard, Emily D. Dolan, Yevgeniy B. Sirotin, Michele Tully, Arthur Wingfield, and Michael J. Kahana. 2006. 'Temporal Associations and Prior-List Intrusions in Free Recall'. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 32 (4): 792–804. doi:10.1037/0278-7393.32.4.792.