

A Probabilistic Agent-Based Simulation for Community Level Language Change in Different Scenarios

Merijn Beeksma*

Hugo de Vos*

Tom Claassen**

Ton Dijkstra* ***

Ans van Kemenade* ***

MERIJN.BEEKSMA@STUDENT.RU.NL

HUGODEVOS.VOS@STUDENT.RU.NL

TOMC@CS.RU.NL

T.DIJKSTRA@DONDEERS.RU.NL

A.V.KEMENADE@LET.RU.NL

*Radboud University Nijmegen, Center for Language Studies (CLS), PO Box 9103, 6500 HD Nijmegen, The Netherlands

**Radboud University Nijmegen, Institute for Computing and Information Sciences

***Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour

****Radboud University Nijmegen, dept. of English Language and Culture

Abstract

We built an agent-based model (ABM) to simulate historical language change, and tested it by means of a case study on word order change in English. Our modeling approach assumes that complex patterns in population-level language change can be understood in terms of many small changes, resulting from interactions between individual agents of different populations. Each agent has a language model that changes due to contact with other agents from the same or the other population. As a result, micro-level changes (i.e. at the level of individual agents) lead to macro-level changes (i.e. at the level of the population). We implemented, manipulated and explored the effect of learning rate, likelihood of interaction between agents from different populations, location-bound dialects, and degree of agent variation within a population. Although parts of the model leave room for fine-tuning, and external factors have yet to be combined into one model, the simulation results show that ABM is a useful tool for gaining more insight into historical language change. The ABM approach has potential for modeling word order change in English, as well as language change in general.

1. Introduction

1.1 Modeling language change

Many internal and external factors may cause a language to change, and change is often the result of an interaction between multiple factors (Campbell 1998). Modeling historical language change may help to uncover its underlying causes. This will not only further our understanding of the change itself, but may also help us to disentangle the complex, dynamic web of social and cultural interactions between members of a population, due to the interaction between language and society.

Sociolinguistic research on language change usually focuses on two dichotomies: the individual versus the group, and the system versus the use of the system (Kerswill 2010, p.1-4). Whether or not language change occurs, and if so, in what way, is to an important extent determined by the type of community. There is more variation at the level of the individual than on the level of the group, and the individual variation depends strongly on the interaction with the individuals' social network (Kerswill 2010, p. 3-4). However, social networks are often very large and complex, and transcend above the social characteristics and geographical location of the individual. The more close-knit a community is, the more change of the language system may be inhibited, and the more the system is held onto by individual members, while looser networks allow and may even stimulate language

change (Milroy and Milroy 1985, Kerswill 2010). Due to such interactions, linguistic variation and change are best considered by taking into account both individual and group characteristics.

The interplay between individuals and the population as a whole led us to explore an agent-based approach for simulating language change. Factors that are known to affect language change can be implemented in agent-based models as parameters that stimulate or inhibit language change. Simulations can then be done to see if the model reproduces historical data. Whenever simulation patterns are similar to the real data patterns, the implemented factors may be factors that played a part in the real-life linguistic change.

In the present paper, we focus on a case study of a historical word order change in English. Between 1400-1650, English changed from a language that made use of verb second, to a language in which verb second is only applied in a limited set of constructions. ‘Verb second’ (v2) describes the phenomenon of constructions in which the finite verb is the second constituent of the sentence, regardless of the type of first constituent. In this study we consider sentences introduced by two different types of adverbial phrases: adverbs of the type ‘then’(THEN) and adverbs of other types (ADVO). Both of these adverb types influence subject-verb inversion. We limit our historical case study to contexts in which the subject of the v2 clauses is a pronoun, in order to narrow down the scope of variation. Thus, we focus on the fluctuations and eventual gradual decrease of the use of v2 in adverb-initial sentences. Examples 1 and 2 show constructions with v2 inversion, which were frequent in the Early Middle English period (1150-1350):

- (1) þa **was** he underfangen mid micel wurtscipe
then was he received with great worship
then he was received with great worship

- (2) nu **shalt** þu hatien þo þe iuel hauen don
now shall you hate those who evil have done
now you are to hate those who have done evil

Towards the Late Middle English period, the number of clauses not associated with the v2 inversion increased, and fluctuations in the use of v2 occurred, eventually leading to a decrease in v2 use. In Modern English, v2 constructions still occur, but only in limited contexts, such as in so called *wh*-phrases and in phrases with an emphatic negative first element. Examples 3 and 4 show v2 inversion in Modern English:

- (3) When **will** he be back?
- (4) At no point **will** he drive a car.

Although the syntactic change has been object of study for years (Van Kemenade 2012), why and how the change occurred is still not fully understood. To clarify the underlying mechanisms, our study has sprung from the research agenda formulated in Van Kemenade (2016) and is exploratory of character. It is intended as a proof of concept, and as a basis for future research in the context of the project initiated by Van Kemenade (2016).

In particular, the goal of our study is to explore the feasibility of agent-based modeling for simulating a complex case of historical language change. The model is applied to data concerning the historical development of v2 in English, to see whether or not agent-based simulations provide insight into historical language change. Based on our current understanding of the change, we assume first that there was initially more variation in the language in general. This was accompanied in some contexts by an increase in systematicity of v2 due to language contact between Anglo-Saxon and Scandinavian speakers following the Viking contest of the Northeast of England in the 9th-10th centuries. Second, we hold that the decline of v2 set in after 1350 and was largely completed by 1500, as a result of dialect contact in the Late Middle English period (c.f. Kroch and Taylor 1997).

In sum, our approach focuses on two interacting populations with different linguistic backgrounds, and simulates different contexts to see how they influence language change. Relevant features are selected based on Van Kemenade (2012, 2016), who investigated different factors of influence on v2 variation. The following sections elaborate on agent-based modeling and our approach towards a better understanding of the historical events that influenced the use of the v2 construction.

1.2 Agent-based modeling

Agent-based models are used to simulate ‘complex systems’, in which elements or agents interact at the level of the individual, but their combined effects change the global representation of a phenomenon (Bonabeau 2002). Furthermore, the agents in a complex system are autonomous operators; they are not controlled by an external force. Generally, complex systems are open to the addition and deletion of agents, as well as the information that is being passed around, thereby inhibiting the system to achieve a steady balance (Steels 1997).

In agent-based models, individual agents are created and made to interact with each other, in order to reconstruct or predict complex social phenomena. Agents are characterized by certain cognitive properties, goals, a position within time and space, and membership of a larger population. Together, agents constitute an interactive network that reflects many individual actions.

Language change results from individual variation, adaptation and self-organisation. Within agent-based models variation leads to change, because the more agents adopt a certain new construction, the more a certain grammatical option has the chance of becoming the new standard, thereby illustrating a positive feedback loop. The process of self-organisation implies that agents tend to adapt to the norm. This approach allows a model to dynamically represent linguistic variation at the level of idiolects, sociolects, dialects and the language as a whole.

Steels and colleagues have been pioneers in developing agent-based models for linguistic research. They used ABMs to explore the origin and evolution of change (Steels 1997, Steels et al. 2002), experiment with language as a self-organizing system, for example as the primary mechanism for vocabulary development (Steels 1995), and explain how very large populations are able to converge on the use of a particular linguistic constructions without global coordination (Baronchelli et al. 2006), to highlight just some of their work. Other examples of applications of multi-agent models can be found in De Bie and De Boer (2007), Lestrade et al. (2016), and Bloem et al. (2015). The model proposed by De Bie and De Boer demonstrates that language change is caused by language mutation. By means of interaction agents adopt these mutations from other agents. In their simulations, De Bie and De Boer show how different language patterns can continue to exist in parallel. Lestrade et al. used an agent-based model to illustrate the plausibility of their Optimality Theory-based hypothesis regarding differential object-marking. Bloem et al. demonstrate that agent-based models are useful tools in research on historical language change, by simulating the historical development of word order in verbal clusters. Their model shows that the present-day word orders in West-Germanic languages are likely to have developed and then deviated from Proto-Germanic word orders. Because this model makes only minimal assumptions, it is well-suited as a basis for our own simulation purposes.

1.3 Properties of the model

Our language simulation model is based on the work of Bloem et al. (2015). Bloem et al. simulate the emergence of languages with ascending and descending word order on a micro (i.e. individual) level, which leads to an accurate simulation of language change at macro (i.e. population) level. The language change at the macro level is assumed to result from the changing language models of all individual speakers of a language.

The model consists of agents that represent the speakers of a language. Like real speakers, the agents are able to ‘interact’. Interaction is represented by one agent producing a sentence and

another agent hearing this sentence. Interactions between agents result in internal changes of the language models of the agents. For example, the probability that an agent produces the same sentence it just heard may increase.

The internal language model plays a key role in the simulation. It determines the probabilities of the types of sentences spoken by the speaker and it changes according to the interactions. The language model consists of a ‘pool’ of exemplars of which one can be selected to create a sentence. These exemplars contain information about the parts of the sentence that are relevant for the simulation. In the case of Bloem et al. these exemplars hold information about:

1. whether the sentence is:
 - (a) a main clause
 - (b) a sub clause
2. whether the auxiliary is:
 - (a) a modal verb + infinitive
 - (b) ‘to have’ + participial main verb
 - (c) a copula + participial main verb

(Scheme adapted from Bloem et al. (2015, p. 24))

The word order of the sentence is determined by the selected exemplar. Bloem et al. compute the probability of each sentence type with Formula 1, in which $p(asc | x_{F1})$ and $p(asc | x_{F2})$ are derived from historical data. We deviate a little from this formula. Instead of accumulating both probabilities, we take the average, as we do not want to end up with probabilities larger than 1. This leads to Formula 2.

$$p(asc | x) = p(asc | x_{F1}) + p(asc | x_{F2}) \quad (1)$$

$$p(V2 | x) = \frac{p(V2 | x_{F1}) + p(V2 | x_{F2})}{2} \quad (2)$$

This model provides a good starting point for simulating language interactions, because it already captures the concepts of agents, language models, and interactions. We adopted these concepts in our own model. Our agents and their interactions function in a similar way as they do in the work of Bloem et al. Additionally, the process of updating the language models following an interaction is conceptually the same. We make the following three assumptions regarding language and indicate how we will implement them.

1. *Assumption*: Language learning is based on imitating others (unconsciously).
 - *Implementation*: Learning is based on the frequency of occurrence of different language varieties. The more often a certain syntactic construction is used among speakers, the more likely other speakers are to learn and use it too.
2. *Assumption*: There are limitations on the variability of an individual’s language model. Although the English language allows freedom in the creation of syntactic constructions for example, not all word orders are considered to be well-formed.
 - *Implementation*: Agents have a binary choice between a v2-construction and a non-v2 construction.

- *Implementation*: Variation between the language models of different agents within the same population is minimal.
3. *Assumption*: Language can be influenced by external factors.
- *Implementation*: There is contact between different populations.
 - *Implementation*: Agents have a predetermined ‘life expectancy’.
 - *Implementation*: Agents are influenced by the norm to a certain degree.

The base model is used to create a baseline. It explores the core effects of interaction between agents over time, as well as the effect of the cognitive language model, the effect of the language norm, and the effect of contact between different populations. The similarities with Bloem et al. only hold for the basic functionalities of the model.

In addition to the concepts introduced by Bloem et al., we also modeled agents’ sensitivity to language change. Agents that have a high ‘preference’ for a certain variant are less likely to be influenced by an agent that uses a different variant, than agents that have no pre-determined preference. Another deviation from the model created by Bloem et al. is that we created multiple populations of agents rather than one. Although these populations are similar of structure to the group of agents created by Bloem et al., the crucial difference is that our agents interact with agents from their own population as well as agents from other populations. We tracked the language change per population instead of the overall language change, although the overall language change can be derived from those.

Furthermore, we extended the model in various ways to explore the effects of multiple internal and external factors that played a role in the language change regarding v2 use. We extended the model with a number of ‘modules’ to simulate different concepts that influence language change. In these modules, some of the assumptions mentioned above were challenged.

The first module we implemented varies the degree of cross-interaction between the populations over time, to simulate how different degrees of contact between the populations influence the rate of adaptation (Section 3.1). The second module simulates the concept of generations. In this module, agents ‘die’ after a certain period of time. They are replaced by new agents that may have a different set of characteristics - they may be more receptive to change of their language model, for example (Section 3.2). The third module simulates more variation between individual agents. In the basic model, the individual variation at the start is minimal. This module is used to determine whether or not more variation between agents at the micro level has an impact on the population at the macro level (Section 3.3). Finally, the fourth module simulates how dialects may arise as a result from geographical separation. This module simulates the ‘real life’ fact that not every agent can interact with every other agent, but only with agents that are located close to the agent itself. This simulation can be seen as a first approximation of a social network.

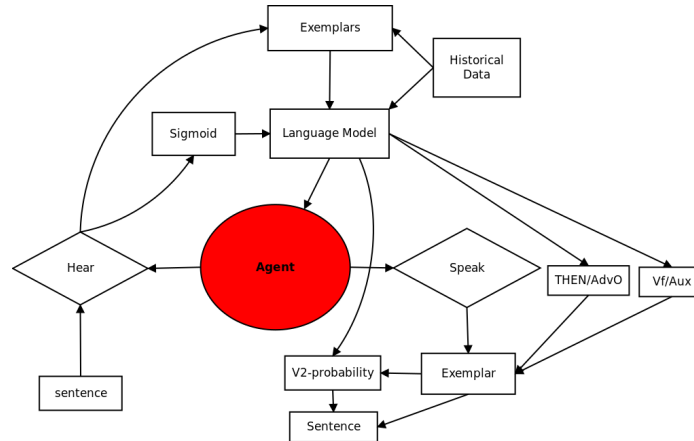
2. The base model

2.1 Description of the base model

The basic principles of our model are that agents represent their population, and that interaction influences an agent’s language model. The agent-based model includes two populations with agents that all have individual language models, which determine the way in which they interact. We first describe the structural qualities of an agent, and then address the way in which agents interact.

A visualization of the model is given in Figure 1. The core of the model is the agent. The most important component of the agent is its language model, which is used to generate utterances. Utterances consist of a) an exemplar and b) an indication of whether or not v2 is ascribed to that exemplar. Whenever an utterance is produced or heard, the language model of the listener is

Figure 1: Schematic representation of an agent.



modified. The language model of the speaker may be modified as well, depending on the settings of the model's parameters.

2.2 Method

The language model

The model¹ contains a collection of exemplars, which are instances of all possible utterances that can be produced with all the permutations of the values for the two features *adverb type* and *verb type*. These exemplars form the basis for the production of an utterance.

To produce an utterance that either has v2 or not, the language model receives information from the exemplar collection. The initial language model is generated according to the historical data so that the proportion of the exemplars is the same as the proportion of the features in the historical data. After its initialization, the collection of exemplars is constantly modified as a result of the interactions.

To determine the verb type and the adverb type, one exemplar is drawn from the exemplar collection. Whether v2 is used is decided on the basis of the selected exemplar. Based on the historical data, the probability of v2 given the exemplar is computed. According to this probability it is determined whether or not v2 is ascribed to the exemplar. The combination of the exemplar and the presence or absence of v2 lead to an utterance which is passed on from speaker to hearer.

Speaking and listening

An agent is capable of performing two actions: speaking and listening. When an agent speaks, it generates an utterance. When an agent has generated an utterance, the exemplar that forms the basis for it, is removed from the language model of the speaker.

When an agent hears an utterance, the exemplar that it contains is added to the agent's language model. As a result, the probability that this agent will produce a similar utterance increases. This way, the utterances that an agent hears influence the utterances that the agent produces.

The interaction procedure

A single interaction in the simulation proceeds as follows. First two agents are selected, a speaker and a listener. The two agents are either members of the same population, or members of different

1. All code that forms the base of this model and the other models in this paper is published on <https://github.com/hdvos/ABM-for-linguistic-change>

Figure 2: Result of the baseline simulation. The y-axis shows the probability of v2 and the x-axis the number of interactions.

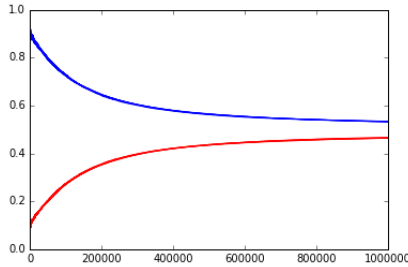


Table 1: Artificial data

Feature 1:	Feature 2:	Population 1		Population2	
verb	verb	<i># v2</i>	<i># total</i>	<i># v2</i>	<i># total</i>
VF	THEN	100	1000	900	1000
VF	ADVO	100	1000	900	1000
AUX	THEN	100	1000	900	1000
AUX	ADVO	100	1000	900	1000
<i>Avg. proportion v2:</i>		<i>10%</i>		<i>90%</i>	

populations. After selection, the speaker speaks a sentence and the listener processes that sentence. This results in a change of their language models.

2.3 Explanation and demonstration of the different components

The following sections will demonstrate the effect of different parameters and modules on language change. The simulations are initially applied to artificial data and then to the historical data. The purpose of this paper is not primarily to make historically accurate predictions, but to provide insight in the operation and effects of different model components.

The general language models for the two populations in the artificial data start at different ends of the spectrum, to magnify the effects of different parameter settings in the simulations. Table 1 shows the language models per population for the artificial data, and Table 2 for the historical data. In the tables, VF stands for ‘finite verb’, AUX for ‘auxiliary verb’, THEN for ‘then-type adverbs’, and ADVO for ‘other adverbs’.

To illustrate the effect of fine-tuning module-specific parameter settings, basic parameter settings are used to demonstrate the modules unless mentioned otherwise (see Table 3 for an overview of the basic parameter settings). The function of most parameters will be introduced throughout the text, at the point where they become relevant. The following sections will clarify parameters that play a role in the base model. Section 3 focuses on the different modules. The result of our baseline simulation is shown in Figure 2. In Figure 2 (and all figures alike), the x-axis represents time in terms of the number of interactions. The y-axis represents the probability that v2 is applied to an utterance. The different colors represent the different populations (blue is Population 1, red is Population 2).

Cross-interaction

The cross-interaction parameter determines the degree to which the two populations interact. A

Table 2: Historical data, period M3

Feature 1:	Feature 2:	Population 1		Population2	
verb	verb	<i># v2</i>	<i># total</i>	<i># v2</i>	<i># total</i>
VF	THEN	190	479	15	36
VF	ADVO	84	640	15	16
AUX	THEN	149	234	56	59
AUX	ADVO	75	283	26	26
<i>Avg. proportion v2:</i>		<i>30%</i>		<i>82%</i>	

Table 3: Parameter settings

Parameter	Setting
Size Population 1	500
Size Population 2	500
Initial Utterances	80
Cross-interaction	0.5
Nr Interactions	1,000,000
Print every n	50,000
Remove Exemplars	True
Doubt step	0
Doubt Influence	0
n runs	5
Use sigmoid	True
Growth factor verb(VF)	0
Growth factor adverb (THEN)	0
Use Death	False
Die after n	n.a.

cross-interaction value of 0.5 means that 50% of all the interactions takes place between agents from different populations. In Figure 3, we present a simulation with a cross-interaction value of 1.0. Compared to Figure 2, both populations converge considerably faster because the mutual influence is larger due to the higher cross-interaction value.

Figure 3: Simulation with a value of 1 for cross interactions.

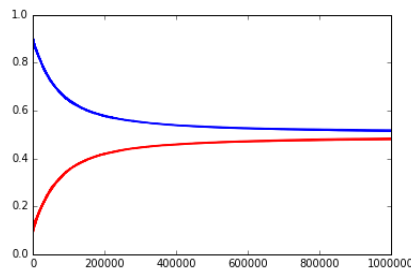
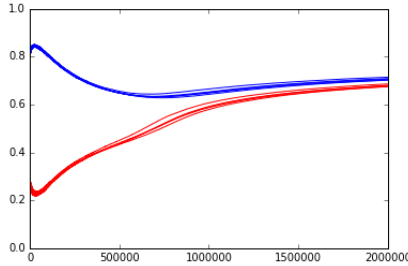


Figure 4: Simulation with the sigmoid formula turned on.



Preference

We modeled preference using a sigmoid formula for the probability of using a v2-sentence as a function of the preference. The function models an agent’s preferences for a certain language variety. This preference is represented as a scale on which every position corresponds with a probability of using v2. A high preference for v2 leads to a high probability of creating v2. Every agent has a position on this scale. Every time an agent produces or hears a v2-sentence, the preference for v2 increases with a fixed amount (determined by the parameter ‘Doubt step’) and every time an agent speaks or hears a non-v2 sentence, it decreases with that same amount. As the relation between the preference and the v2-probability is logistic, the v2 probability changes more when an agent has no specific preferences. However, when an agent already has a high preference for (or against) v2, the v2-probability changes considerably less.

In the language model, a weighted average is computed between the v2-probability based on the historical data and the v2-probability based on the sigmoid formula. The weight of the sigmoid-based probability is set with the parameter ‘Doubt influence’. The simulation presented in Figure 4 is executed with a doubt step value of 2, and a doubt influence value of 2. This figure shows that both populations converge with each other, but not to 50% as happens in previously seen other figures. This finding reflects that agents without preference ($p(v2) = 0.5$) are more susceptible to change than agents with a clear preference.

Growth factors

We implemented so called growth factors (Bloem et al. 2015) into the model, to simulate the influence of the norm, i.e. the language variant that is most accepted by the community. In reality, speech is not the only modality through which language is transmitted. Language variation is often limited by written language. While language variation could be limitless in theory, in reality usually only a couple of competing variants exist. In pursuit of a uniform system, regulations and strong preferences often impose a norm that is accepted by a large part of the population. Although deviating from the norm in general does not cause problems, and because speech is generally more variable than written text, the norm is not necessarily inviolable. People however tend to be conscious of the golden standard, and therefore tend to conform their language use to the norm.

Within the model, agents delete exemplars from their memory once they uttered them. However, when growth factors are applied, not all exemplars are deleted. For each feature, the norm is determined, e.g. the norm for verbs is verbs of the type ‘finite’, and for adverbs the norm is adverbs of the type ‘other’. This would lead to the growth factors g_{VF} and g_{ADVO} . Each growth factor additionally receives a value x . When a growth factor applies, utterances that contain the feature specified by the growth factor are *not* deleted from the speakers memory once every x occurrences. So, if g_{VF} is set to 5 for example, once in every five occurrences that an utterance including a verb of the type ‘finite’ is uttered, the exemplar is *not* removed from the speaker’s language model. Adjusting x to be a higher or lower value simulates the degree to which agents obey the norm. Not

Table 4: Timeline with fictional data. The interaction numbers represent moments in time. The varying cross-interaction values are the corresponding events.

Interaction number	Degree of cross-interaction
0	0
250	0.5
500	0.8
750	0.1

removing exemplars results in a higher probability of the feature being included in new utterances, thereby preserving the norm.

3. Modules extending the base model

In addition to the base model, we implemented four modules. Each module is implemented separately from the other modules, allowing us to study the effects of each module and its specific parameters in isolation. Section 3.1 illustrates how a varying amount of contact between populations influences the course of the language change. The module described in Section 3.2 simulates how different generations of agents influence the rate of adaptation. Section 3.3 shows whether or not a higher degree of individual variance influences the population at the macro level. Finally, Section 3.4 addresses the effect of geographical separation. All demonstrations make use of the parameter settings as listed in Table 3, unless stated otherwise.

3.1 Time-dependent cross-interaction

The amount of cross-interaction between different populations within the base model is implemented as a steady factor. There is either no cross-interaction between populations, or there is to a certain (fixed) degree. While this would be a realistic scenario in a situation in which two populations live next to each other, for example in case of neighboring countries, or in case of groups with different dialects within a country, it is not a realistic scenario in our case.

Before our populations came into contact with each other, a situation existed in which there was no language contact at all between the populations. This period was followed by an episode of intense contact. It is very plausible however, that during this period the degree to which the populations cross-interacted was subject to fluctuation. When the populations parted again, they did not do so completely. The influence that both populations had on each other remained, and changes as a result of language contact kept occurring even though no intense cross-interaction took place anymore. However, the language change was of a different nature than when the populations were still interacting intensively. The module described in this section aims at simulating a varying degree of contact between populations.

Method

In contrast to the base model, in which the cross-interaction variable is set to a constant value, this module allows us to determine the amount of cross-interaction per moment in time. When we run a simulation, we can define one or more points in time and determine the degree of cross-interaction we want to take place from that moment on. Time is represented in the model by the accumulating number of interactions. Table 4 therefore shows a ‘time line’: the left column shows the number of interactions, i.e. the moments in time, and the right column shows the changes in the degree of cross-interaction between the populations.

The example in Table 4 simulates a situation in which there is no language contact at all between the two populations at first. Then, after 250 interactions, the groups come into contact and

Figure 5: Varying levels of cross-interaction between the populations. The dotted lines mark the starting points of a new cross-interaction value. The cross-interaction values are shown for each interval at the top of the figure. The *left* figure shows a simulation with one moment of contact between the two populations. In the figure on the *right* a simulation is shown where two populations get in contact with each other after a while, followed by a period without contact, which is followed by a period in which they have intense contact.

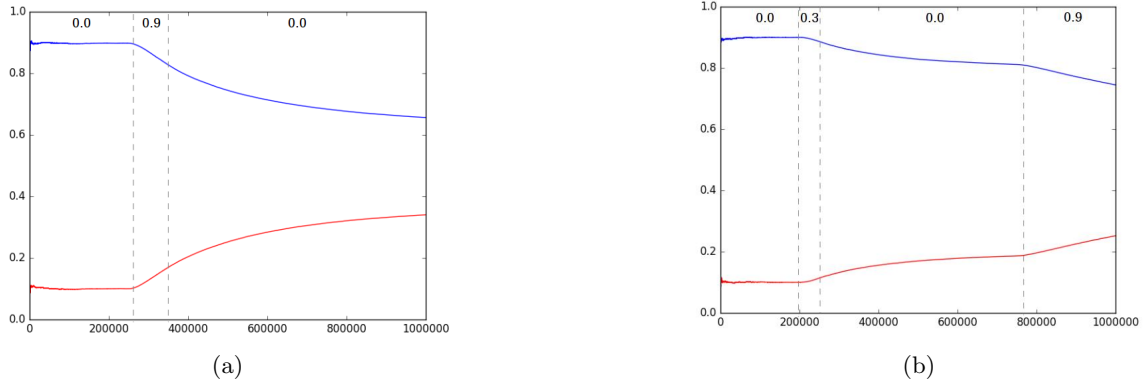
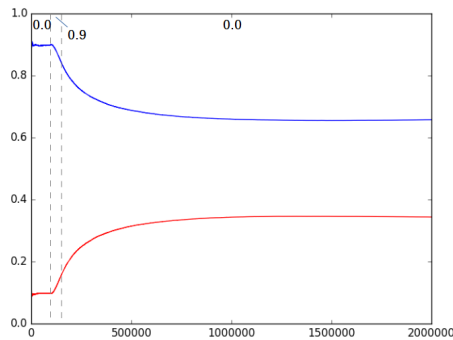


Figure 6: The use of v_2 stabilizes gradually after contact between the populations stops. The dotted lines mark the starting points of a new cross-interaction value. The cross-interaction values are shown for each interval at the top of the figure.



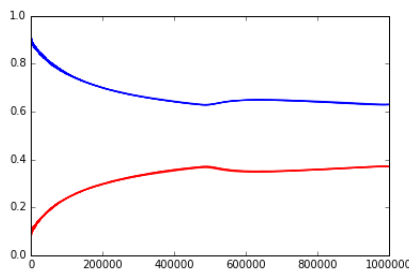
start interacting. After 500 interactions the contact intensifies, and finally after 750 interactions the contact between the populations minimizes.

Results

Figure 5 illustrates two examples of the result of applying this module. The dotted lines indicate the moments in time at which a change in the degree of cross-interaction between the populations occurs. Both parts of Figure 5 show that when the populations stop cross-interacting, their language models gradually stabilize again but on a different level than where they started, as we would expect. However, because this process takes time, the figures do not show fully stabilized models. Figure 6 shows that with more time, the use of v_2 does stabilize fully.

This module enables us to vary the degree to which populations interact, and the figures above show that such variation leads to the expected results: the populations converge faster when a higher level of cross-interaction is applied, and slower when lower levels of cross-interaction are applied. Short moments of intense contact between populations may lead to definitive changes for both

Figure 7: Result of the simulation when parameter ‘Die after n’ is turned on.



populations. Although the levels of v2 use of both populations stabilize after a moment of contact, they do not return to their original state. With varying degrees of cross-variation the populations keep showing the tendency to converge, but the course of change deviates from the course of change shown by the base model.

3.2 Generations

The generations module provides the system with new agents that may have new characteristics. These new agents replace old agents from the same population one by one. To fully understand how the generations work, we will first introduce a concept we call *death*, which is essential for the generations module. After that, we will explain the generations in detail.

Death

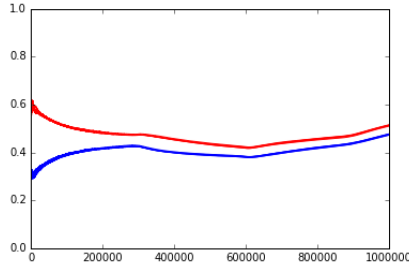
‘Death’ is the process in which an agent is removed from the simulation and replaced by a new agent. For every agent individually, the model tracks how often they have participated in an interaction (either as speaker or as listener). This amount can be set by the parameter ‘Die after n’. After an agent participated in n interactions, it is removed from the simulation. A new agent is added to the population of the removed agent to keep the number of agents equal through the whole simulation, and therefore keep the population sizes stable. In this version of the model, the new agent is initialized according to the data with which the other agents were initially loaded (but the agent could be initialized with new data as well).

Compared to the parameter settings described in Table 3 we changed the value for ‘Use Death’ to True and the value for ‘Die after n’ to 1000, which means that every agent dies after 1000 interactions.

The results of these simulations are shown in Figure 7. The figure shows a wave-like pattern occurring at about 500,000 interactions. This is caused by the fact that at this point most agents die. Because agents participate in interactions at random, at 500,000 interactions roughly 500 agents have had 1000 interactions and are therefore removed all at roughly the same moment. The wave-like pattern is the result of the assumption that newly generated agents make use of the initial language model based on the input data. The new agents are conservative and therefore delay the language change.

One would expect new agents in a population to be very open to change however, in the way that L1 learners learn a language as it is when they are learning it, rather than the way it was when their parents were born. The function of this module, however, is not (yet) to realistically model the death and birth of speakers, but to show the possibility and effect of gradually replacing agents in a population. One of the applications of the concept of death is in the generations module, which will be described in the next part.

Figure 8: Effect of different generations with different language models.



Generations

The idea behind *Generations* is that dying agents are replaced by agents who possess different language models. This contrasts with the process of death describe above, where agents are replaced by agents that have the same initial language model as the agent they replace. The start of a new generation is when the language model of all the new agents is different, compared to the initial agents language models. In the current implementation of this module, whenever a new generation starts, a new set of historical data is loaded. This is the only change in the language model. In future implementations, other elements of the model could be varied between generations as well. For example, the parameters of the sigmoid formula or the cross-interaction value can be varied upon to represent a changed attitude towards a second language.

In Figure 8, a simulation is shown with artificial data. The artificial data set for this simulation is different from the data shown in Table 1. There are two arguments for this: the first is that the original artificial data exists for only one generation; the second is that we wanted to enhance the underlying notion of ‘generations’.

The input data for the *Generations* module is different from the data in the rest of the simulations. The reasons for this are, first, that the artificial data in the rest of the model only have one generation, and second, that we exaggerated the differences between the generations to illustrate the concept behind generations.

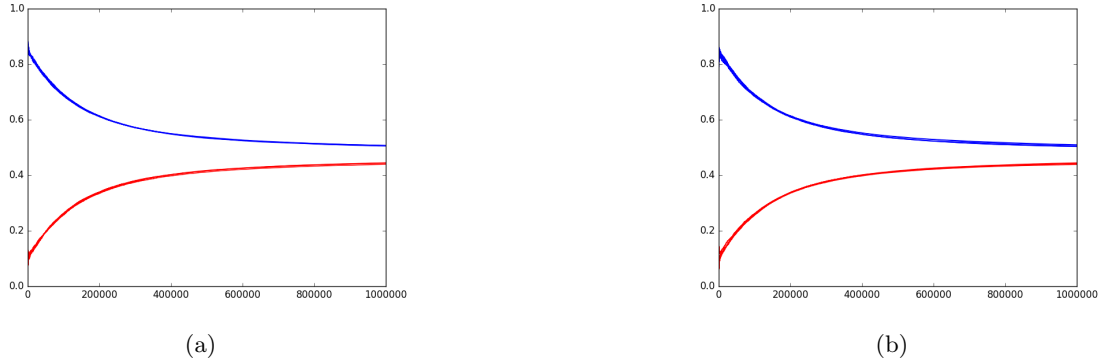
In this simulation, a new generation is set to start at 30% and at 60% of the number of interactions. The effect of a new generation only shows whenever new agents are born. At 30% and 60%, a ‘bump’ is visible as a result of the start of a new generation. At about 90%, a third ‘bump’ is visible. This is just the result of a regular death and birth pattern as described in the previous section.

As the figure shows, every time a new generation starts, a new impulse is given to the language change. There is a clear effect of the new generation on the whole process. The implementation is not perfect though. First of all, death needs to be more sophisticated, so that not all agents die at about the same time. Secondly, this module gives opportunities for introducing more new elements in the model than just a different language model. For example, new agents can have different settings for the sigmoid formula, or as is the case in Section 3.1, a different value for cross-interaction.

3.3 Individual variation

This module aims to simulate the fact that different individuals have similar but slightly different language models: it simulates idiolects. Contrary to the other simulations that assume that all agents have almost identical language models, this simulation allows for much more individual variation. Therefore, instead of all agents being similar instances of one ‘super agent’, we create a continuum of agents that all differ from each other to a certain degree.

Figure 9: Simulation on artificial data (see Table 1) with standard parameter settings (see Table 3). In the simulation on the *left* no individual variation is applied, while in the simulation on the *right* there is. There is no visible difference between the results.



Method

We added a parameter ‘v2 noise’, which receives a value between 0 and 1. This value determines the amount of variation we want to ascribe to each individual agent. The value varies the historical data for each agent, from which the agents’ exemplars are generated.

The total number of sentences per feature in the historical data is multiplied with the v2 noise value. The outcome determines the degree to which the number of v2 sentences may vary per agent. We will illustrate this with an example.

We set the value of v2 noise to 0.025, and apply it to the feature combination VF THEN of Population 1 (number of v2 sentences = 100, total number of sentences = 1000, see Table 1). To determine the outer boundaries of the variation of v2 for each agent for this feature combination, the value of v2 noise is multiplied by the total number of sentences: $0.025 * 1000 = 25$. The outcome is then applied to the v2 count. The range in which the number of v2 sentences must fall for each agent is now as follows: 100 ± 25 , so [75 – 125].

For each agent, the number of v2 sentences is determined to be a random point in that range, i.e. one agent’s language model may consist of 89 v2 sentences out of 1000 sentences, while another agent’s model may consist of 113 v2 sentences out of 1000. Because the points in the range are determined randomly, the points will be evenly distributed around the mean of the population.

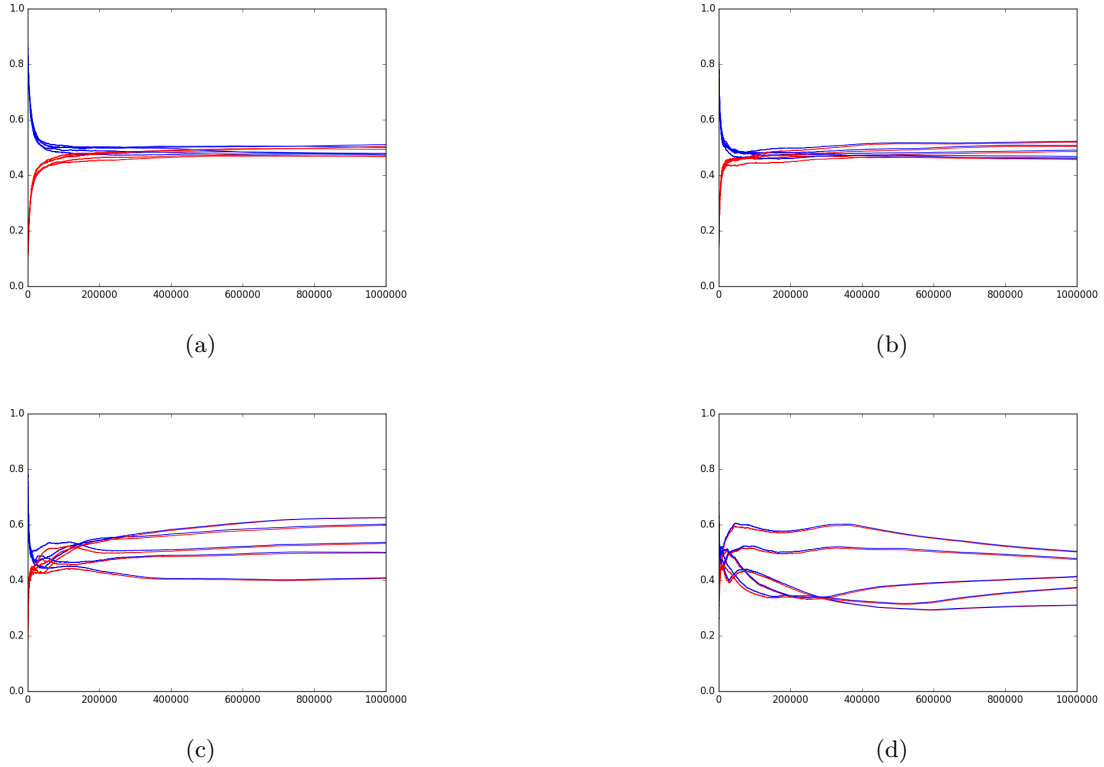
Results

The data from which the exemplars are generated, is varied to a certain degree for each agent, instead of being exactly the same for all agents, allowing for individual variation. The simulation is designed in such a way that the variation applied to the agent’s language models is evenly distributed around the mean, so at the level of the populations, the mean does not change, as is illustrated by Figure 9.

On the individual level, however, the results depend strongly on the altered historical data, caused by the v2 noise parameter. When instead of 500 agents per population, we only consider very small groups, we can observe a larger degree of variation. The smaller the groups are, the larger the variation is. See Figure 10.

Figure 11 illustrates the effect of a varying v2 noise value. The more variation is allowed in creating the agent’s language models, the more variation we see in the model. In Figure 11, the variation is limited to 10%, because both populations have rather extreme starting values of v2 use: Population 1 uses it in 10% of the cases, and Population 2 in 90% of the cases. Because we want the variation to be equally distributed around the mean, the maximum variation that can be applied to this artificial data is 10%.

Figure 10: Simulation on artificial data (see Table 1) with standard parameter settings (see Table 3), except for the population sizes, which are varied. These figures show the effect of a v_2 noise value of 0.025 (2.5%) on populations of varying sizes (size $n = 25$, $n = 10$, $n = 5$, and $n = 2$, respectively).



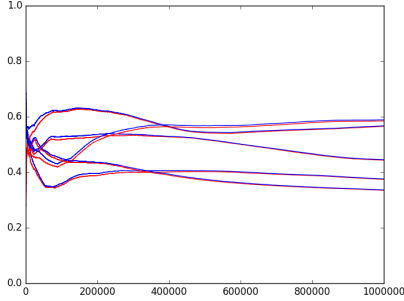
Figures 10 and 11 show that allowing individual variation leads to different results at each run. Another striking characteristic of adding noise is that the course of change fluctuates much more than without noise. These two factors make the course of change at the individual level much less predictable, but they do so without changing the overall patterns at the macro level of the population. The module simulates the fact that although individuals may develop in different, somewhat unpredictable ways, the population as a whole develops in a more consistent way, and is not affected by individual fluctuations.

3.4 Locations

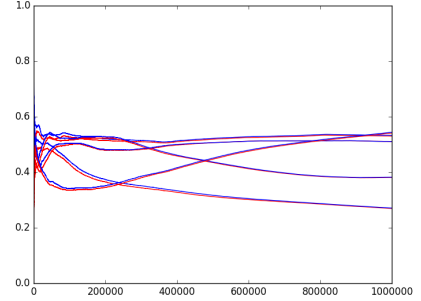
In the base model, agents that participate in an interaction are chosen at random, which results in the possibility for each agent to interact with any other agent. In reality, however, not all members of a population interact with all other members, as they live in different geographical areas. Moreover, in a situation in which two populations cross-interact, it is likely that only certain members of both populations (that work in the same profession, for example), would communicate with each other.

The third simulation is set up to simulate the fact that not all members of a population interact with each other, and, in case of cross-interaction, that not all members of one population come in contact with all members of the other population. Simulating subpopulations which are separated by geographical boundaries, leads to variations in the course of language change. The resulting

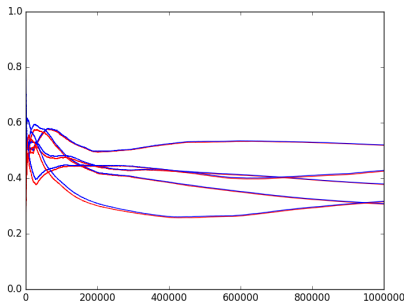
Figure 11: Simulation on artificial data (see Table 1) with standard parameter settings (see Table 3), except for the population sizes, which are set to $n = 2$ for each figure. Figures a-d show the effect of a varying v_2 noise value, with 1%, 2.5%, 5%, and 10% variation allowed, respectively.



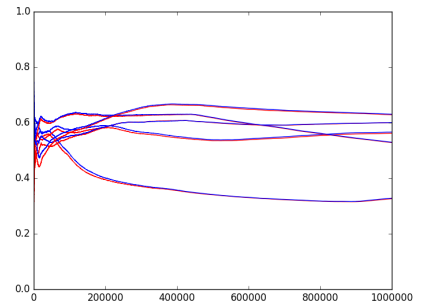
(a)



(b)



(c)



(d)

variations may be considered as dialects.

Method

This module allows us to divide the populations over a number of locations. The allocation of agents to locations happens for each population separately. Therefore, geographical areas of different populations may overlap either entirely, to a certain degree, or not at all. The distribution of agents over the locations is done manually by determining the number of agents for each location per population, therefore the distribution of agents over locations is not necessarily even.

An example is shown in Figure 12. The figure shows two populations, both consisting of 500 agents, distributed over five locations. Each population is distributed over three locations. The populations meet in the middle; they overlap only in location 3.

In the simulation, different parameters are used to determine the following factors:

- the number of locations;
- the distribution of agents over the locations per populations (and thereby the degree of overlap between different populations);
- the extent to which agents are allowed to communicate with agents of a different population. By default they are only allowed to interact with agents in their own or an adjacent location.

Figure 12: Distribution of agents over locations per population. The example makes use of the parameter settings in Table 3. Blue dots represent agents from Population 1, red dots represent agents from Population 2. The arrows show which interactions between locations are allowed according to the default settings.

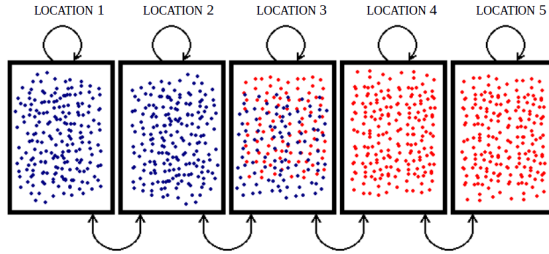


Table 5: Distribution of agents over locations per population.

	Location 1	Location 2	Location 3	Location 4	Location 5
Population 1	200	200	100	–	–
Population 2	–	–	100	200	200
<i>Total</i>	<i>200</i>	<i>200</i>	<i>200</i>	<i>200</i>	<i>200</i>

Results

In order to demonstrate the variability of the outcome, the simulation is presented here with different parameter settings, and with the artificial data as described in Table 1.

Figure 13 shows the results of the example described in Figure 12. As can be seen in Table 5, each population is divided over three locations. Locations 1 and 2 both contain 200 agents from Population 1, location 3 contains 100 agents from both populations each, and locations 4 and 5 both contain 200 agents from Population 2. The distribution of the populations over the locations is therefore symmetrical in this example.

The figure illustrates that the course of the language change depends on the location. The change is most prevalent in location 3, which contains both populations, because the amount of cross-interaction is the highest in this location. The adjacent locations show a less rapid change, but more rapid than locations 1 and 5. Per location, locations 2 and 4 consist of agents of only one population. They are, however, able to communicate with agents from adjacent locations, and therefore the two dialects spoken in these locations are still subject to change.

Per location, locations 1 and 5 also consist of agents of only one population. These locations are not adjacent to locations containing agents from the other population. Their language model, however, still is susceptible to change, because of the change experienced by the other members of

Figure 13: Simulation of two cross-interacting dummy populations distributed over five locations. The distribution of agents over locations is given in Table 5.

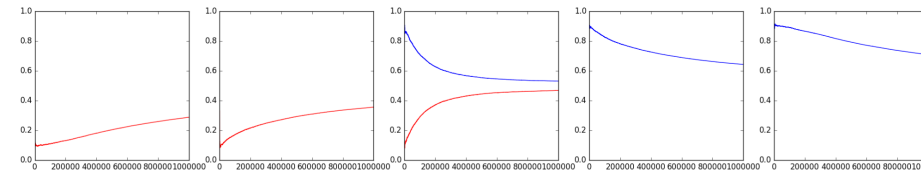
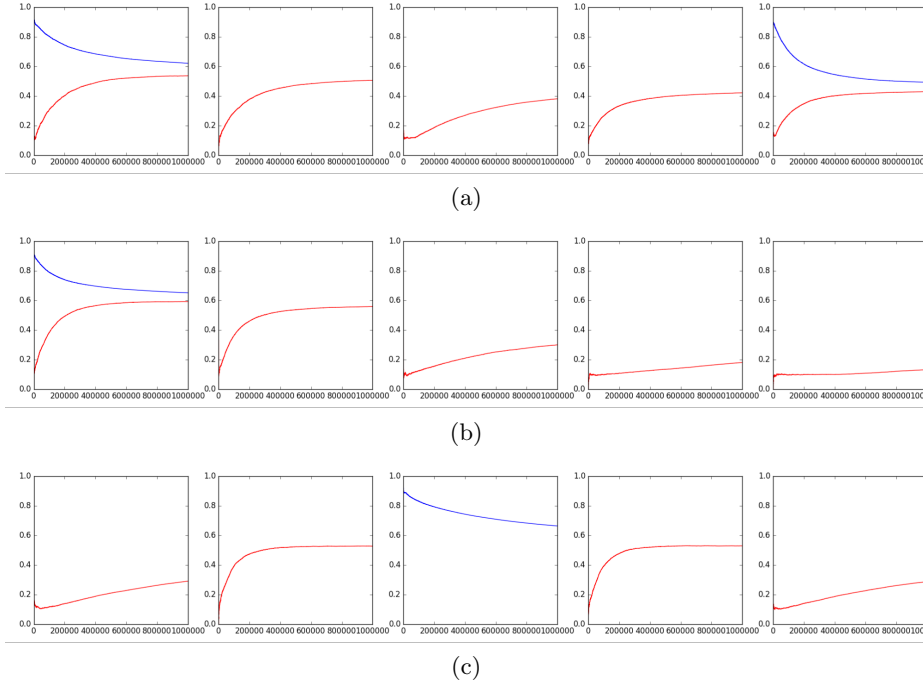


Table 6: Distribution of agents over locations per population (artificial data, see Table 3).

		Location 1	Location 2	Location 3	Location 4	Location 5
Figure 14 (upper)	Population 1	50	75	100	125	150
	Population 2	250	—	—	—	250
	<i>Total</i>	<i>300</i>	<i>75</i>	<i>100</i>	<i>125</i>	<i>400</i>
Figure 14 (middle)	Population 1	100	100	100	100	100
	Population 2	500	—	—	—	—
	<i>Total</i>	<i>600</i>	<i>100</i>	<i>100</i>	<i>100</i>	<i>100</i>
Figure 14 (lower)	Population 1	200	50	—	50	200
	Population 2	—	—	500	—	—
	<i>Total</i>	<i>200</i>	<i>40</i>	<i>500</i>	<i>50</i>	<i>200</i>

Figure 14: Three variations (one per row) on the data shown in Figure 13. Simulation of two cross-interacting dummy populations distributed over five locations. For each of the figures, the distribution of agents over locations is given in Table 6.



their population in the adjacent location. The further away from the cross-interaction a location is, the less influenced it is by the cross-interaction, thereby leading to dialects.

Simulations do not need to be symmetrical. Table 6 and Figure 14 illustrate different scenarios. Figure 14 (upper) shows a symmetrical distribution of agents over locations, but with cross-interaction only in the outer locations. Figure 14 (middle) shows the effect of one population distributed evenly over all five locations, cross-interacting with another population only in one location. Figure 14 (lower) illustrates a situation in which agents from different populations never cohabit in the same location, but only in adjacent locations.

This module allows us to implement geographical distribution and separation of populations. The examples shown in Figure 14 illustrate that different distributions of agents over populations leads to different patterns of change. Depending on the number of locations, the distributions of populations over locations, the degree to which the populations cross-interact, and the degree to which agents can communicate with agents residing in other locations, influence the course of change to a large degree. Once we have more knowledge of these types of information in the historical texts, we can use this model to simulate the historical context in detail.

4. Conclusion

In this paper, we formulated an agent-based model for simulating language change. We illustrated the potential effects of several factors that presumably influenced the historical development of v2 in English.

Our model consists of a base model and a number of isolated modules simulating individual factors that may play a role in language change. The results of the base model simulations and the four modules show that changes at the micro-level (i.e. the level of the individual) indeed lead to changes at the macro-level (i.e. the level of the population). However, the module in which individual variation is applied (section 3.3) illustrates that the larger picture is not necessarily affected by noise on the micro level. While individuals do influence the overall model, the population as a whole can nullify individual variation. As a consequence, the overall population changes only when individuals vary systematically.

The simulations discussed in this paper represent only a part of the model's ability to model historical language change. Exploring the entire range of parameter settings and the influence of input data is a challenge that is beyond the scope of this paper. The simulations do, however, illustrate the model's potential and feasibility: even though the isolated modules are not yet implemented into one collective body, they already show interesting results. The model seems to be able to reproduce complex patterns found in historical data, and combining model features may lead to more accurate and complex reproductions of historical events and the ensuing language changes.

Because of the flexibility of the model, we believe it has the potential of developing into an accurate model for simulating language change. Moreover, we believe that the complex patterns created by this relatively simple model are encouraging. It was our aim to show that agent-based modeling offers a unique angle for language change research and is a useful tool for gaining insight into the complex interactions of factors involved in the course of language change. Therefore, we regard this pilot study as promising for further exploration.

5. Future work

One of the most important tasks in future research is testing the model against empirical data. Our model was intended as a proof of concept, and as a basis for future research in the context of the project initiated by Van Kemenade (2016). Although the current model successfully displays change in agent behavior as a direct results of tuning the parameters and thereby allowing the recreation of historical events, we consider it to be among the first steps in a line of research concerning agent-based models for modeling linguistic change.

Many model processes allow for improvement and refinement. Due to the time constraints, some assumptions were implemented rather simplistically. For instance, with respect to the 'death' procedure, we assumed that every agent dies after exactly n interactions. This could be interpreted as every agent dying at exactly the same age. This assumption about death could be nuanced in future versions of the model. A simple solution would be to have agents of different ages in the starting conditions of the model.

A different example involves the concept of generations, and concerns a more fine-grained introduction of new agents. Instead of basing the composition of a new generation on different historical data alone, other parameter settings could be varied for each new generation as well, such as a different cross-interaction value, a more sensitive sigmoid formula, or a more strict or loose acceptance of the norm, by applying the growth factors differently. This way, a new generation becomes a tool for ‘injecting’ new types of agents into the model.

In addition to tweaking implementations of factors that may have influenced the course of change historically, future work should also focus on mimicking historical events such as epidemics or mass migration. Besides macro-level changes, changes in the architecture of individual agents can be implemented. The agent’s ‘age’ could be used to influence the learning rate for example, to simulate differences in learning abilities of children and adults. Similarly, agents may become more or less recipient of the norms during their lifetime. Finally, an obvious step in further development is to combine various modules into one model to create a holistic simulation of reality.

Apart from improvements in the computer model, a proper reproduction of the course of change requires further analysis of the historical data. The amount of available data for our study was limited. More detailed input information improves the initiation of the model and the accuracy of the simulations, but also allows more extensive model evaluation. Finally, a more thorough analysis of the historical data may lead to the identification of more relevant features than verb type and adverb type alone. (Lestrade et al. 2016)

References

- Baronchelli, Andrea, Maddalena Felici, Vittorio Loreto, Emanuele Caglioti, and Luc Steels (2006), Sharp transition towards shared vocabularies in multi-agent systems, *Journal of Statistical Mechanics: Theory and Experiment* **2006** (06), pp. P06014, IOP Publishing.
- Bie, Pieter de and Bart de Boer (2007), An agent-based model of linguistic diversity, *Language, Games, and Evolution* p. 1.
- Bloem, Jelke, Arjen Versloot, and Fred Weerman (2015), An agent-based model of a historical word order change, *Conference on Empirical Methods in Natural Language Processing*, p. 22.
- Bonabeau, Eric (2002), Agent-based modeling: Methods and techniques for simulating human systems, *Proceedings of the National Academy of Sciences* **99** (suppl 3), pp. 7280–7287, National Acad Sciences.
- Campbell, Lyle (1998), *Hystorical Linguistics: An introduction*, MIT Press, Cambridge, Massachusetts.
- Kemenade, Ans van (2012), *Sage handbook of sociolinguistics*, Sage Publications, London, chapter Sociolinguistic approaches to language change: *phonology.*, pp. 219–235.
- Kemenade, Ans van (2016), Word order change online: Language change, second language acquisition, computational modelling and simulation.
- Kerswill, Paul (2010), Sociolinguistic approaches to language change: phonology, *Sage handbook of sociolinguistics* pp. 219–235.
- Kroch, Anthony and Ann Taylor (1997), *Parameters of Morphosyntactic Change*, Cambridge University Press, Cambridge, UK, chapter Verb movement in old and middle English: Dialect variation and language contact., pp. 219–235.
- Lestrade, Sander, Geertje van Bergen, and Peter de Swart (2016), *Optimality-Theoretic Syntax, Semantics, and Pragmatics. From Uni- to Bidirectional Optimization*, Oxford University Press, Oxford, UK, chapter On the origin of constraints, pp. 179–199.

- Milroy, James and Lesley Milroy (1985), *Authority in language: Investigating language prescription and standardisation*, Routledge.
- Steels, Luc (1995), A self-organizing spatial vocabulary, *Artificial life* **2** (3), pp. 319–332, MIT Press.
- Steels, Luc (1997), The synthetic modeling of language origins, *Evolution of communication* **1** (1), pp. 1–34, John Benjamins Publishing Company.
- Steels, Luc, Frédéric Kaplan, Angus McIntyre, and Joris Van Looveren (2002), Crucial factors in the origins of word-meaning, *The transition to language* **2** (1), pp. 4–2, Oxford Press, UK, to be published.