Computational Linguistics in the Netherlands Journal 1 (2011) 21-40 Submitted 7/2011; Published 12/2011

Language Change and SA-OT. The case of sentential negation

Alessandro Lopopolo A.Lopopolo@student.uva.nl
Tamás Biró T.S.Biro@uva.nl
ACLC, University of Amsterdam

Abstract

Simulated Annealing for Optimality Theory (SA-OT) updates Optimality Theory by adding a model of performance to a theory of linguistic competence. Our aim is to show that SA-OT can contribute to language change simulations. Performance "errors" are considered to be one of the causes of variation and change. We have chosen to model the evolution of sentential negation (SN). The descriptive background adopts *Jespersen's Cycle*, according to which the evolution of sentential negation follows three main stages (1. pre-verbal, 2. discontinuous, and 3. post-verbal). Therefore, we advance a novel model for SN, based on SA-OT. It reproduces the three pure and the two observed mixed stages, whereas it correctly predicts the lack of an intermediate stage between 3 and 1. The success of the approach corroborates the computational, performance-based approach to the data. Finally, we employ the *iterated learning* paradigm to reproduce historical changes in a "simulated corpus study". This enterprise turns out to be more difficult than one would naively believe.

1. Introduction

Linguistic systems change over time, this is a well-known fact. Many theoretical attempts have been laid down in order to explain the process of change. This paper discusses the role of imperfect mental computation ("performance errors") in the history of one particular syntactic phenomenon, sentential negation. For that purpose, we employ *Simulated Annealing for Optimality Theory*, a recently developed computational implementation of Optimality Theory (Biró 2006, 2005, 2009). Our aim is in fact twofold: On one hand, to study the evolution of sentential negation, taking as starting point both the seminal works of Jespersen (1909, 1917) and Dahl (1979) and the recent optimal theoretical analysis advanced by de Swart (2010). On the other hand, we want to test SA-OT as a computational model of not only linguistic performance, but also language variation and change.

This paper is structured in six principal sections. Section 2 will introduce Simulated Annealing for Optimality Theory, comparing it to traditional OT and how it can incorporate performance. Section 3 will then introduce our case study, sentential negation (SN), Jespersen's historical stages, and de Swart's approach. In this section we also raise some criticism against her analysis, arguing that SA-OT may do better than traditional and Stochastic OT in reproducing language typology and historical change. Section 4 will outline our model and introduce the candidate set, the topology, the set of constraints and the hierarchies employed in the simulation. Section 5 will describe the results of these simulations, comparing them with Jespersen's observed or postulated stages and de Swart's account thereof. Section 6 turns to the dynamics driving the change, presenting the results of a multiagent *iterated learning* experiment. Finally, section 7 will conclude the paper.

2. Simulated Annealing for Optimality Theory (SA-OT)

In order to understand what SA-OT is and how it handles variation, we compare it to traditional OT (Prince and Smolensky 1993/2004). We assume that competence and performance are two distinct concepts (Chomsky 1965), one represented by a grammar (in our case, a set of ranked constraints) and the other being its implementation (Smolensky and Legendre 2006, Bíró 2006). Traditional OT is a theory of grammar, determining what forms are *grammatical*: those that are optimal for a list of ranked constraints. SA-OT, an implementation of OT, is an algorithm that searches for these best candidates, but may fail to find them. Thereby, it predicts the forms *produced*, including "performance errors". The term 'error' refers to anything that is ungrammatical with respect to the grammar, but still produced: fast speech forms, acceptable irregular forms and other variations. SA-OT does not aim at accounting for *all* types of variation, as small random divergences are sometimes better reproduced by other stochastic variants of OT (Boersma 1997).

More specifically, a grammatical form is a *global optimum*, *i.e.*, a candidate that optimizes a harmony function (specified by the constraint ranking) on the set of *all* possible candidates. At the same time, a produced form may be both a global optimum, but also a *local optimum* that is globally not optimal: a candidate that is more harmonic than its *neighbors*, as we shall explain soon.

A grammar is thus a *harmony function* H over a set of possible candidates $\{w, w', \dots\}$. It is composed of elementary functions C_i called constraints ($0 \leq i \leq N$). A constraint assigns a number of violation marks to the candidates according to certain requirements (avoid a structure, similarity to input, etc.). Moreover, the constraints are ranked into a language specific hierarchy:

$$C_N \gg C_{N-1} \gg \dots \gg C_0 \tag{1}$$

In turn, the harmony function assigns a vector, called a *violation profile*, to each candidate w, consisting of the violation marks assigned by the constraints:

$$H(w) = (C_N(w), C_{N-1}(w), \dots C_0(w)) \tag{2}$$

The grammar determines the candidate that maximizes harmony. Maximization of harmony corresponds to minimizing the number of violation marks, at least for the higher ranked constraints. Candidate w_1 is *more harmonic than* candidate w_2 if and only if $H(w_1)$ is lesser than $H(w_2)$ by the lexicographic order. In other words, we first seek the *fatal constraint*, that is, the highest ranked constraint that assigns a different number of violation marks to the two candidates. Then, the candidate that suits this constraint better is the more harmonic candidate with respect to the hierarchy (to the grammar). Optimality Theory postulates that the most harmonic candidate in the entire candidate set, the *global optimum*, is also the grammatical form.

The *Simulated Annealing for Optimality Theory Algorithm* (SA-OT) attempts to find this global optimum, but sometimes fails to do so. A *topology* (or *neighborhood structure*) is introduced, on which SA-OT performs a random walk. The topology is defined on the search space, the OT candidate set, usually by neighborhood criteria called *basic steps*. It

is the 'horizontal component' of the landscape in which the random walk takes place. The 'vertical component' is provided by the harmony function, and thus, the random walk turns into hill climbing. The random walk starts from an initial candidate w_{init} in the search space. At each iteration step, it proceeds by choosing a random neighbor w' of its current position w. Whether the random walker actually moves from w to w' is governed by a *transition probability*, which we return to in a moment. Initially, the random walker is free to move anywhere; later, it will only move to more harmonic neighbors. The random walk terminates in a *local optimum*, a candidate that is more harmonic than its neighbors, and this form is finally returned by the algorithm. Consequently, SA-OT, as a model of performance, predicts that not only the global optimum, but also further local optima are uttered by speakers. It also predicts their frequencies.

At any moment of the algorithm, the *transition probability* depends on w and w' (in fact, only on the 'difference' of $H(w')$ and $H(w)$); as well as on the parameter *temperature*, a pair of numbers $\langle K, t\rangle$, which decreases following a cooling schedule. Let us compare w to w' in the way it is usually done in OT, and let us identify the fatal constraint F. Let d be the difference in violations of the fatal constraint: $d = F(w') - F(w)$. If d is negative, then w' is more harmonic than w. Additionally, let f denote the *rank* (or, rather, the *K-value*) of F: a value associated to each constraint, which is higher if the constraint is ranked higher in the hierarchy.

Then, SA-OT defines the transition probability – the chance of the random walker actually moving from w to the randomly chosen neighbor w' – as

$$P\big(w \to w' | \langle K, t\rangle\big) = \begin{cases} 1 & \text{if } w' \text{ is not less harmonic than } w, \text{ else} \\ 1 & \text{if } f < K \\ \exp(-d/t) & \text{if } f = K \\ 0 & \text{if } f > K \end{cases} \tag{3}$$

In other words, if the randomly chosen neighbor w' is more harmonic than (or equally harmonic to) the current position w of the random walker, then the new position becomes w'. Otherwise, let us compare the rank (K-value) f of the fatal constraint to the first component K of the temperature. If K is larger, the random walker moves to w'. If f is larger, the random walker stays in w. Finally, if $f = K$, then a random number r is generated with a uniform distribution on the $[0, 1]$ interval, and if $r < \exp(-d/t)$, then the random walker moves to w'.

Thus, we approach variation through performance. SA-OT maintains the traditional dichotomy between competence and performance. Competence is modeled by the set of universal constraints, their language specific ranking and the candidate set. Performance emerges from the topology and the random walk heuristic, which will or will not return the grammatical candidate. For background and details of SA-OT, please refer to previous papers of the second author, as well as to the Appendix (pseudo-code and parameters). The concrete case in sections 4 and 5 will further illustrate the content of this probably abstract introductory section.

3. Case study: Sentential Negation

Sentential negation (SN), as it is considered here, is the possibility to reverse the truth condition of the main verb in the sentence. The study of the ways languages mark this function dates back at least to the Danish linguist Otto Jespersen (1909, 1917), who observed three main types of SN: pre-verbal, discontinuous and post-verbal sentential negations. These types are also argued to be three historical stages, in this diachronic order. Later, Dahl (1979) coined the term *Jespersen's Cycle* to describe the apparently cyclical nature of the evolution of SN.

3.1 Types of sentential negation

Languages such as Italian, Chinese, Russian and Hungarian (examples 1) mark sentential negation pre-verbally. The language specific sentential negator (*non, bu, ne, nem*) is placed on the left side of the main verb, appearing in a pre-verbal position in the linear order of the constituents. Conversely, languages such as Lombard, Dutch, Turkish, and Japanese (examples 2) mark sentential negation in a post-verbal position, the sentential negator (*mia, niet, -me-, -na-*) being placed on the right side of the main verb (verbal root in Turkish and Japanese). [1]

Example 1

a. *Giovanni non mangia la mela.* [Italian]
Giovanni SN eats the apple.
'Giovanni does not eat the apple.'

b. *tā bu sǐ.* [Chinese]
3sg SN die.
'S/he won't die.'

c. *Katja ne čitaet knigu.* [Russian]
Katja SN reads book.ACC.
'Katja does not read the book.'

d. *János nem alsz-ik.* [Hungarian]
János SN sleep-3sg.
'János does not sleep.'

Example 2

a. *Giovanni al maja mia la mèla.* [Eastern Lombard]
Giovanni CL eats SN the apple.
'Giovanni does not eat the apple.'

b. *Jan eet de appel niet.* [Dutch]
Jan eats the apple SN.
'Jan does not eat the apple.'

1. The Italian, Lombard and French examples are compiled by the first author, the rest is borrowed from de Swart (2010).

c. *John elmalar-i ser-me-di-ø.* [Turkish]
John apples-ACC like-SN-past3sg.
'John didn't like apples.'

d. *Taroo-wa asagohan-o tabe-na-katta.* [Japanese]
Taroo-TOP breakfast-ACC eat-SN-past.
'Taroo didn't eat breakfast.'

Discontinuous sentential negation is the third possible type observed by Jespersen. It consists of two negators, one positioned on the left, and the other on the right of the main verb, yielding a negator-verb-negator linear sequence. French, Cairese Piedmontese, Old English and Welsh are among the languages that employ this type of sentential negation (examples 3).

Example 3

a. *Jean ne parle pas anglais.* [French]
Jean SN speaks SN English.
'Jean does not speak English.'

b. *U n li sent nent.* [Cairese Piedmontese]
3.CL SN him hears SN.
'He can't hear him.'

c. *Ne bið he na geriht.* [Old English]
SN is he SN righted.
'He is not forgiven.'

d. *Doedd Gwyn ddim yn cysgu.* [informal Welsh]
SN.be.impf.3sg Gwyn SN PROG sleep.
'Gwyn was not sleeping.'

Jespersen noted that these three types of sentential negation often represent three evolutional stages in the history of many European languages. He pointed out that pre-verbal sentential negation was often replaced by discontinuous negation, which, in turn, developed into post-verbal SN. This is particularly evident looking at the history of French and English. The table below, based on de Swart (2010:104), sums up the diachronic succession of the three stages. Post-verbal sentential negation in French corresponds to contemporary colloquial French, while in English, it represents Early Modern English.

	pre-verbal	discontinuous	post-verbal
French	Jeo ne dis	Je ne dis pas	Je dis pas
English	Ic ne secge	Ic ne seye not	I say not
	1. *SN V*	2. *SN V SN*	3. *V SN*

Beside these three stages, it is important to add that some languages represent mixed stages, where two types of sentential negation are simultaneously produced by speakers.

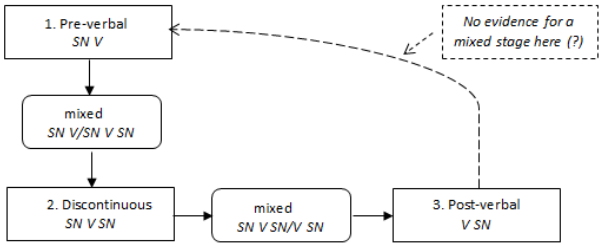

Figure 1: Jespersen's cycle: three pure stages and two attested mixed stages. The third mixed stage and the corresponding transition from post-verbal to pre-verbal, although questionable, are widely assumed.

Probably best-known is contemporary French, with both discontinuous ("*ne dis pas*" – *SN V SN*) and post-verbal negation ("*dis pas*" – *V SN*).

Languages with patterns from both stage 1 and stage 2 (cf. Figure 1) are believed to be in a diachronic process moving away from the pre-verbal to the discontinuous stage. A similar story applies to languages, such as contemporary French, that can express negation both in a discontinuous and a post-verbal fashion, and which may adopt a purely post-verbal pattern in the future. Finally, post-verbal SN is hypothesized to evolve into pre-verbal SN, closing thereby *Jespersen's cycle*.

Interestingly enough, there is no strong evidence for a mixed stage between post-verbal and pre-verbal sentential negation; nor for a transition from stage 3 to stage 1. De Swart provides a couple of examples, though. Among them the fact that, with the rise of the *do*-support, the negative marker may be reanalyzed as pre-verbal in present day English. The Shakespearean corpus testifies to the mixed phase. Another example is the fact that *pas* is placed before the main verb in some French-based creole languages. However, these cases are quite controversial proofs for the transition from post-verbal to pre-verbal sentential negation. In fact, the English negator still goes after the inflected verb (auxiliary, dummy *do*, or copular *be*), and a French-based creole language cannot be considered as the next step in the organic evolution of some variety of French. Thus, the lack of proof for the third mixed stage – or for a post-verbal to pre-verbal transition – undermines the very cyclical nature of what has been traditionally termed Jespersen's *cycle*.

3.2 De Swart's analysis employing traditional and Stochastic OT

De Swart (2010) introduces three constraints also used in our SA-OT model (cf. section 4.2, and the explanatory tableau there): *Neg, NegFirst, and FocusLast. Constraint *Neg prefers candidates with less SN. Constraints NegFirst and FocusLast require an SN to occur before and after the verb, respectively. She proposes to link each stage (pre-verbal, discontinuous and post-verbal) to possible OT grammars expressed as rankings of these constraints. Ad-

ditionally, she accounts for the grammar change following Jespersen's cycle by changing the ranking of two neighboring constraints at each stage. Her analysis can be summarized thus:

Stage 1: pre-verbal	1.1	*Neg ≫ NegFirst ≫ FocusLast
	1.2	NegFirst ≫ *Neg ≫ FocusLast
Stage 2: discontinuous	2.1	NegFirst ≫ FocusLast ≫ *Neg
	2.2	FocusLast ≫ NegFirst ≫ *Neg
Stage 3: post-verbal	3.1	FocusLast ≫ *Neg ≫ NegFirst
	3.2	*Neg ≫ FocusLast ≫ NegFirst

The six hierarchies will also reappear in our simulations, although corresponding sometimes to different languages. In de Swart's model, each pure stage can be equally represented by two hierarchies, without any visible difference in the language production. For instance, both hierarchies 1.1 and 1.2 lead to a pre-verbal sentential negation type. Historical change is accounted for by a series of constraint rerankings, and mixed stages are modeled using Stochastic OT (Boersma 1997). For instance, when *Neg and FocusLast are just being switched between 1.2 and 2.1 – and ranked very close, having overlapping noise distributions – the stochastic mixture of the two hierarchies yields both forms.

The symmetry characterizing de Swart's approach predicts that the transition from the post-verbal stage to the pre-verbal one is exactly as simple as the other two transitions (compare hierarchy 3.2 to hierarchy 1.1.). Moreover, that languages in the mixed stage between post-verbal and pre-verbal are just as frequent among the languages of the world as are languages in the other two mixed stages. Thus, the cycle would be indeed closed – as suggested by so many linguists, but which seems to be supported by so few, if any, empirical data.

4. An SA-OT model

Therefore, we introduce a novel model to account for the observations. Being based on an OT framework, our approach also requires the basic components of OT: a candidate set and constraints ranked into hierarchies. Additionally, we will introduce a neighborhood structure (topology) to implement the SA-OT Algorithm.

4.1 Candidates

In our infinite candidate set, a candidate is a pair of underlying form and surface form (*uf*, *sf*). The *uf* represents the semantics, namely, the polarity of the utterance to be expressed; hence, it can be either negative or positive. The candidate's *sf* is a binary syntactic tree, including the main verb (V) and zero or more sentential negation markers (SN). We have left out all other possible sentence constituents, such as arguments (subject, object) and modifiers, since our goal is to focus on the bare expression of negation in the main clause. Figure 2 contains some examples of candidate surface forms.

4.2 Constraints

An OT system also requires a set of constraints that build up the harmony function to be applied on the candidates. Traditionally, there exist two categories of constraints: faithfulness

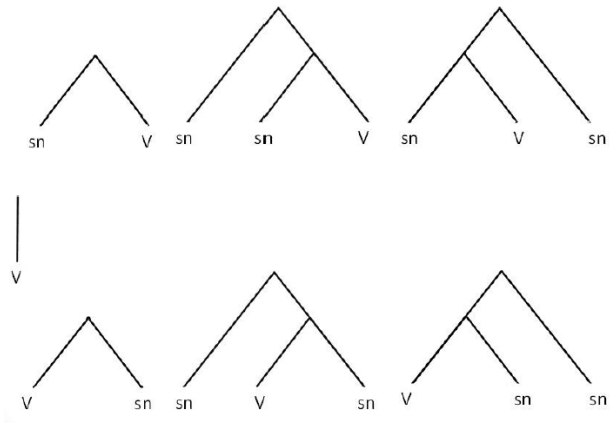

Figure 2: A few surface forms from the infinite candidate set of our model: binary trees with one verb and zero or more sentence negators, placed in pre-verbal and/or post-verbal positions. All other constituents of the sentence are omitted for the sake of simplicity.

and markedness constraints. Faithfulness constraints check whether the output matches certain features of the input (here *sf* and *uf*). Our faithfulness constraint is Faith[Neg]. Markedness constraints, on the other hand, punish candidates that display a certain feature in their *sf*. Our markedness constraints are *Neg, NegLast, and NegFirst. This set of four constraints is directly based on de Swart (2010; but note that we renamed FocusLast as NegLast):

- Faith[Neg]: The polarity expressed by the *uf* must match the presence (for negative polarity) or absence (for positive polarity) of SN in the *sf*. The constraint assigns one violation mark in the case of mismatch.
- *Neg: It punishes any occurrence of SN in the *sf*. It assigns a number of violation marks equal to the number of SN leaves in the surface form.
- NegFirst: It assigns one violation mark to candidates without an SN in pre-verbal position.
- NegLast (FocusLast in de Swart): It assigns one violation mark to candidates without an SN in post-verbal position.

We only used negative polarity as input. Let us have a look at what happens there. Internal parsing brackets are omitted in the following (unranked) tableau, because candidates with the same linear structure but different parses are assigned the same number of violation marks by all four constraints:

/pol = neg/	Faith[Neg]	*Neg	NegFirst	NegLast
[V]	*		*	*
[SN V]		*		*
[V SN]		*	*	
[SN V SN]		**		
[V SN SN]		**	*	
[SN SN V]		**		*
[SN V SN SN]		***		
...				

As an example, consider candidates [V] and [SN V SN SN]. Candidate [V] is assigned one violation mark by Faith[Neg], since it does not display a negative marker, while the input (/pol = neg/) requires it. For the same reason, no violation mark is assigned by *Neg. Yet, the candidate incurs the violation of both NegFirst and NegLast because it does not express sentential negation either in a pre-verbal, or in a post-verbal position. Conversely, candidate [SN V SN SN] is assigned no violation of Faith[Neg], because its surface form matches the input polarity. However, it gets three marks from *Neg, as a consequence of its three sentential negators. Constraints NegFirst and NegLast are simultaneously satisfied by this candidate, because [SN V SN SN] contains both pre-verbal and post-verbal markers.

4.3 Topology

The topology of our model is built on the candidate set described above. Each candidate is connected to its neighbors on the basis of similarities at the *sf* level. The neighborhood of a candidate is defined by referring to simple transformational rules, called *basic steps*. Our model employed the following basic steps:

- Add an uppermost layer with an SN to the left.
- Add an uppermost layer with an SN to the right.
- Remove the uppermost layer.
- Reverse the linear order of the daughters of some node.

Figure 3 displays a small portion – the candidates known from Figure 2 – of the infinite neighborhood structure employed in our model. For instance, the neighborhood of candidate [SN V] is composed of candidates [V], [V SN], [SN [SN V]], and [[SN V] SN], as a result of applying the following steps:

- [SN [SN V]]: add SN marker to the left of the topnode of [SN V].
- [[SN V] SN]: add SN marker to the right of the topnode of [SN V].
- [V]: remove topmost SN marker from [SN V].
- [V SN]: reverse the linear order of the daughters of the top node in [SN V].

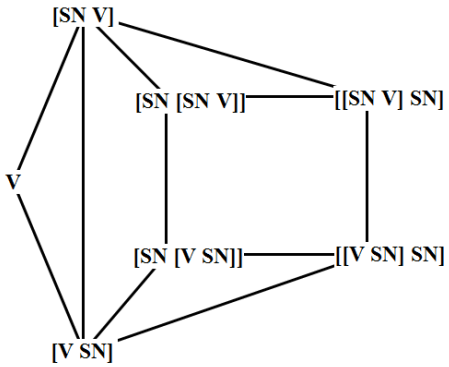

Figure 3: A small portion of the neighborhood structure of the model, displaying the surface forms on Figure 2. The edges of the graph connect the neighbors. Two candidates are neighbors if the *sf* of the one can be transformed into the *sf* of the other in a single step.

4.4 Hierarchies

The above mentioned constraints are ranked in a hierarchy. Each constraint is assigned a ranking value (see also the Appendix). The higher the constraint rank, the more costly its violation. We kept Faith[Neg] fixed at the highest position (rank value 4) and played around with the remaining three constraints. What we obtain are the six hierarchies already listed by de Swart:

Hierarchy 1	Faith[Neg] ≫ *Neg ≫ NegFirst ≫ NegLast
Hierarchy 2	Faith[Neg] ≫ NegFirst ≫ *Neg ≫ NegLast
Hierarchy 3	Faith[Neg] ≫ NegFirst ≫ NegLast ≫ *Neg
Hierarchy 4	Faith[Neg] ≫ NegLast ≫ NegFirst ≫ *Neg
Hierarchy 5	Faith[Neg] ≫ NegLast ≫ *Neg ≫ NegFirst
Hierarchy 6	Faith[Neg] ≫ *Neg ≫ NegLast ≫ NegFirst

5. Experiments: Various grammars and performance patterns

Now, each of the six hierarchies are applied to the neighborhood structure (topology), evaluating the candidates. Thus, we obtain six landscapes on which the SA-OT algorithm performs hill climbing in search for the optima.

The details of our simulations, the pseudo-code of the SA-OT Algorithm, as well as the parameter settings are given in the Appendix. In what follows, we first discuss the six landscapes in a "pen-and-paper" fashion; the predicted qualitative performance patterns will have been confirmed by the computer experiments. Subsequently, we turn to the most interesting case, the mixed stages, and present the quantitative results obtained by using the *OTKit* software package (Biró 2010).

5.1 Hierarchies and optima

Figure 4 reproduces the most interesting subset of the topology, already presented in Figure 3. Arrows have been added that point to the more harmonic one of the two neighboring candidates, with respect to each of the six hierarchies discussed above. Hierarchies 3 and 4 have been reproduced on the same graph, since they only differ in how [V SN] relates to [SN V]. These graphs help us find the local optima for each hierarchy. The reader is invited to check that no candidate with three or more SN leaves can ever be locally optimal.

Hierarchies 1 and 6 rank constraint *Neg above NegFirst and NegLast. They yield a single local optimum, which is also globally optimal: candidate [SN V] with pre-verbal negation, in the case of Hierarchy 1, and its mirror image [V SN], in the case of Hierarchy 6. We predict, and experiments confirm, that SA-OT will produce these forms exclusively: the performance pattern corresponds to the grammatical judgments, because there are no other local optima, which could emerge as eventual performance errors.

Hierarchies 3 and 4 demote *Neg below NegFirst and NegLast, and thereby they yield the discontinuous negation forms ([SN [V SN]] and [[SN V] SN]) as equally most harmonic.[2] Our prediction is that both candidates will emerge in the output of SA-OT, as both are local optima. Experiments show that Hierarchy 3 slightly prefers [[SN V] SN], whereas Hierarchy 4 returns [SN [V SN]] a little bit more often, due to the asymmetry of [SN V] and [V SN]. Moreover, the exact frequencies of the two forms slightly depend on the parameters of the algorithm, as well. Note, however, that candidates [[SN V] SN] and [SN [V SN]] correspond to the same overt form "SN V SN", and we have no means of differentiating between a population producing [[SN V] SN] more often than [SN [V SN]] from a population producing these two forms with a reversed preference. Therefore, we conclude that both Hierarchies 3 and 4 correspond to the languages with discontinuous negation.

Thus far, the SA-OT model runs parallel to de Swart's model. Yet, Hierarchies 2 and 5 behave differently. In traditional OT, adopted by de Swart, these grammars return their global optima, [SN V] and [V SN], respectively. However, SA-OT also returns local optima (as "performance errors"). Observe that the last two landscapes include globally non-optimal local optima: candidate [SN [V SN]] for Hierarchy 2, and [[SN V] SN] for Hierarchy 5. Therefore, we predict that these hierarchies correspond to languages in mixed stages. The exact proportion of the discontinuous forms in the performance pattern can be determined by computer experiments only, and this is the issue to which we turn next.

To sum up, our model correctly reproduced the three pure stages and the two observable mixed stages. There is no room, however, for a third mixed stage (between post-verbal and pre-verbal), which was predicted by de Swart's traditional OT approach, and which have not been observed in the historical data. Moreover, a mixed stage corresponds to a separate grammar in our approach, and not to a stochastic mixture of two grammars. The presence of two forms in the population is not (only) due to the simultaneous presence of 'conservative speakers' and 'innovative speakers'; nor (only) to single speakers entertaining two registers (for instance, a colloquial grammar and a formal one) in their head. But the same grammar may produce both forms, because the computational implementation of the grammar in the speakers' head will also return local optima.

2. In a more elaborate grammar, further constraints – which prefer, for instance, left or right branching structures – might choose between these two candidates.

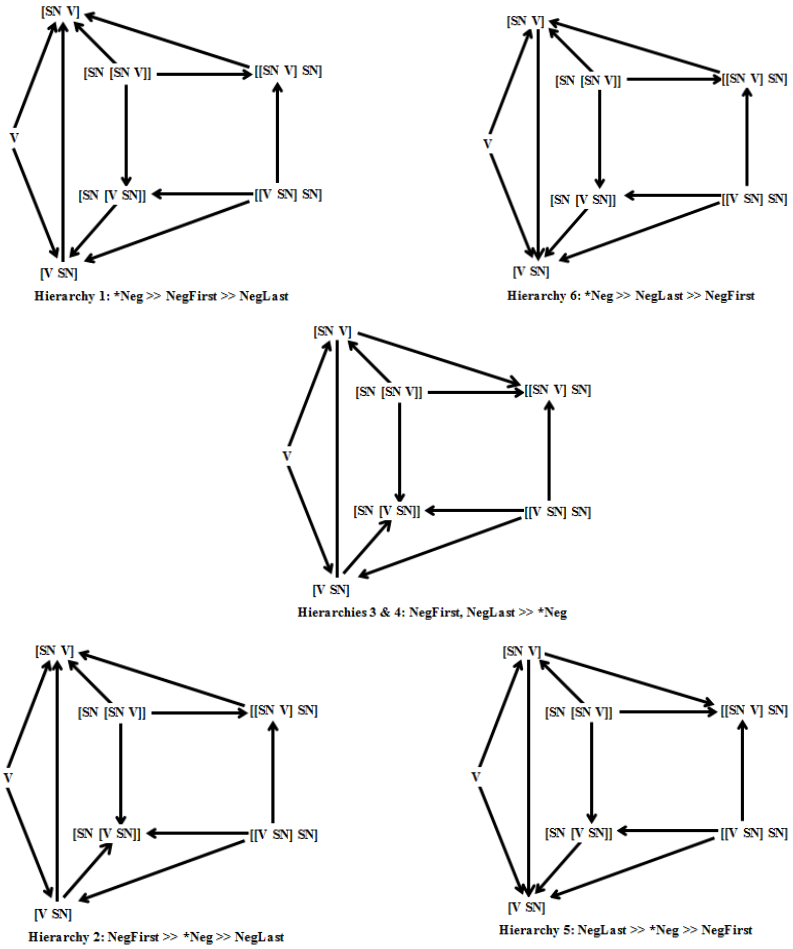

Figure 4: Arrows pointing to the more harmonic form on the topology for each hierarchy. Hierarchies 3 and 4 are combined into a single directed graph (middle panel), as they only differ in the relative ranking of [SN V] and [V SN].

5.2 Production in the mixed stages

Hierarchy 2, which we focus on now, introduces both a global optimum (the pre-verbal negation [SN V]) and another local optimum (form [SN [V SN]] with discontinuous negation). Hierarchy 5 corresponds to a mirrored story – due to the symmetry observable both in the candidate set and in the constraint set – and therefore does not require separate treatment.

Simulated annealing applied to Hierarchy 2 produces the global optimum with frequency p, and the other local optimum with frequency $1 - p$. If we call the global optimum 'grammatical form', and other local optima 'performance errors', then p is the *precision* of SA-OT: the probability of finding the grammatical form. The exact value of p depends on the parameters of the algorithm, and can only be determined with computer experiments. If historical

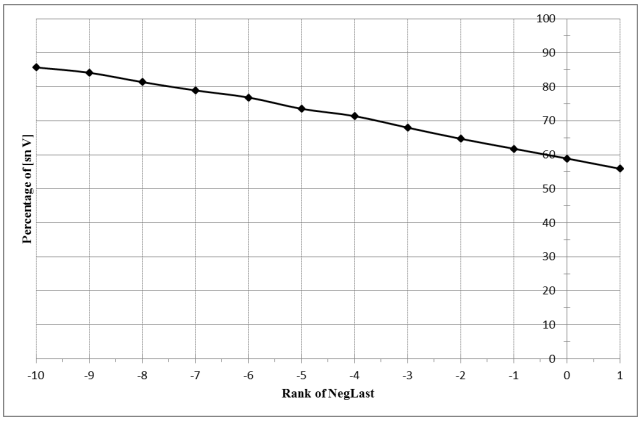

Figure 5: Diminishing the rank (K-value) of the lowest constraint NegLast increases the frequency of the pre-verbal form [SN V] in the performance pattern of Hierarchy 2.

change in Jespersen's Cycle from pre-verbal to discontinuous negation goes via the mixed stage Hierarchy 2, then it is crucial to understand how to fine-tune the frequency p.

The current model behaves in a novel way, if contrasted to past work on SA-OT. As discussed in the Appendix, changing the parameters of the algorithm (`t_step`, `K_max`) does not cause the model to output the global optimum [SN V] with a very different frequency. It is another factor, previously hardly investigated,[3] that makes it possible to create systems with p changing between slightly more than 50% and almost 100%: decreasing the rank (and K-value) of the lowest ranked constraint NegLast increases the probability p of producing [SN V] (see Figure 5).

From the point of view of traditional OT, decreasing the rank of the lowest ranked constraint does not change the grammar: the order of the constraints stays the same, and the harmony of the candidates are also unaffected. And yet, the performance pattern is modified. Namely, increasing the distance in rank (in K-value) between *Neg and NegLast increases the number of iteration steps during which the random walker still can escape from the local optimum [SN [V SN]] to its neighbor [SN [SN V]]; therefrom it may be trapped by the (by then inescapable) global optimum [SN V] (either directly, or via [[SN V] SN]). Thus, a larger difference in rank (in K-value) between *Neg and NegLast enhances the chances of the random walker to end up in [SN V].

Unlike past SA-OT analyses of various phenomena, our current model resembles Stochastic Optimality Theory (Boersma 1997, Boersma and Hayes 2001) in that the frequencies of the different forms (almost) directly "correlate" with the ranks of the constraints. Consequently, learning from data with specific frequencies becomes feasible, and this is where we continue in the next section.

3. With the exception of Bíró (2006), section 7.1.4.

5.3 From individual performance patterns to population frequencies

The model correctly reproduces the 3+2 stages observed by the historical linguists, and predicts the lack of the third mixed stage. It can also mimic a graded shift in frequency in the mixed stages. Yet, a single individual with a mental grammar corresponding to the mixed stage is predicted not to produce discontinuous negation with a frequency higher than 50%.

How can, then, our model reproduce the typical S-shaped curves observed in linguistic changes? (See, for instance, Niyogi (2006), pp. 23-25.) The answer provided in the next section is that the frequency of the novel form *on a population level* will nevertheless follow an approximately S-shaped curve, as the population contains more and more agents with a purely discontinuous grammar. Chains of generations of simulated agents will acquire their grammar (competence) by being exposed to the performance of the immediately preceding generation, before their own performance patterns are recorded for the "simulated historical corpus".

6. Simulating gradual transition from one pure stage to another

On the basis of the grammars sketched above, we developed an *iterated learning simulation* (Kirby and Hurford 2002) in order to test the learning dynamics, and in particular, the transition from one pure stage to another. A population or generation of speakers was composed of five agents. An agent was equipped with an OT grammar (the model of its competence), an SA-OT production procedure (performance) and a learning procedure. The latter used the *Gradual Learning Algorithm* (GLA) of Boersma (1997), with learning plasticity 0.1, but without evaluation noise added to the ranks. After being "born" with a random grammar,[4] each agent was exposed to 300 pieces of learning data produced by the previous generation: each time, a randomly chosen 'adult agent' generated an utterance with underlying form /negative polarity/, which was compared to the production of the learner. Both adults and learners used SA-OT to generate forms. For the sake of simplicity, we ignored eventual social structure and learning-from-peer effects.

During GLA learning, the agent updated its constraint ranking values in order to obtain a grammar whose production was as close to the one of the previous generation as possible. Since the learning input represented only a portion of the production of the previous generation, and this production might contain a percentage of "performance errors", the grammars developed by the learning agents were expected to differ from the one of the previous generation. A second reason for language change is imperfect learning: some learners may not have reached the target grammar by the end of the learning phase. This can be due to a number of reasons, again: if the learner's initial grammar was very different from the target, then the amount of learning data might have been insufficient, but GLA is also known not to converge under every condition (Pater 2008).

Generation 0 was set up with five agents, each with the purely pre-verbal negation grammar (Hierarchy 1), and started "teaching" the newborn Generation 1. When this generation had "grown adult", that is, they had been exposed to 300 cases of learning data, then this

4. Constraint Faith[Neg] was assigned rank 4.9, and the markedness constraints were associated with a random floating point value between -0.1 and 4.9. The standard parameters of the SA-OT Algorithm (`K_max` = 5, `K_step` = 1, `t_step` = 1, etc.) were used, as discussed in the Appendix.

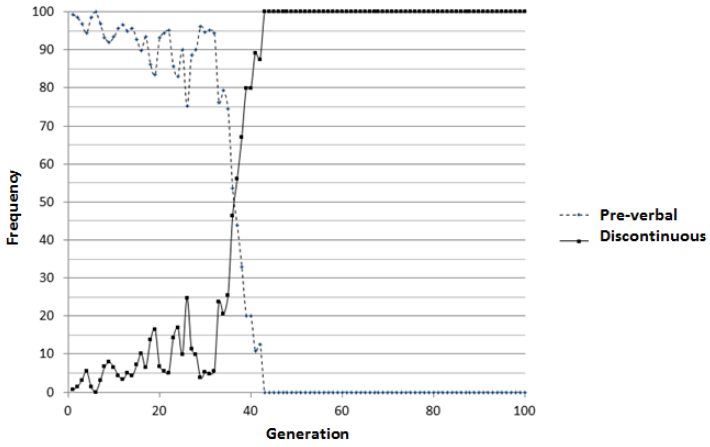

Figure 6: An example of the dynamics of change from pre-verbal to discontinuous sentential negation in a population, during 100 generations. Each data point corresponds to the frequency of a form in the sample 'recorded' by that generation.

generation recorded a production sample of size 500 for the "simulated corpus study". Then, Generation 2 was born, and began to learn from Generation 1, etc. This *iterated learning* ran for a total of 100 generations, and the whole procedure was repeated 20 times. Figure 6 displays the dynamics of one run of the experiment. On average, the process of learning from performance leads to a gradual shift from a pre-verbal to a discontinuous pure stage, as predicted above. Notice the S-shaped curve on the population level.

To our surprise, we have observed that the pre-verbal pattern is highly unstable, and the system rapidly moves to the discontinuous stage. Clearly, the numerous parameters of this highly abstract model need to be refined, and/or further factors must be taken into account, in order to reproduce the languages that steadily employ pre-verbal negation for a longer period in their history. At the same time, populations with a discontinuous negation language are very stable. As a consequence, the language community did not replace discontinuous with post-verbal SN, and hence, we have been unable to reproduce the whole history of English and French. A more careful analysis of the details of the model is deferred to future work; nevertheless, we are optimistic about the reproducibility of history.

The twenty experiments contained one hundred generations each, yielding the "simulated corpora" of 2000 (non-independent) populations. The histogram in Figure 7 displays the distribution of these samples. Observe the clusters towards the higher end of the histogram. They are due to the conspiracy of two factors: the small population size (five agents per generation) and the lack in our model of grammars yielding the discontinuous form with a frequency between 50% and 100%. In turn, populations of five agents with a purely discontinuous grammar will produce a sample with 100% of discontinuous forms (removed from the histogram, as they proliferate among the 2000 populations), whereas

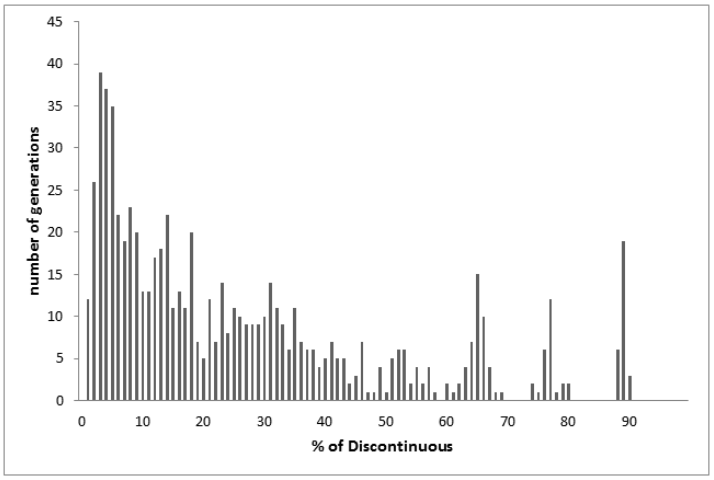

Figure 7: Number of generations producing specific percentages of discontinuous negation.

populations with four such agents only cannot produce a "corpus" with more than 90% of discontinuous negations. Hence, we do not expect any data point between 90% and 100%. The peak just below 90% corresponds to the generations in which four of the five agents have acquired a purely discontinuous grammar, and one agent has a mixed grammar, but the constraints ranked such that it will produce the local optimum [SN [V SN]] in almost half of the cases. The second peak from the right, just below 80%, corresponds to two such agents, joining three purely discontinuously negating speakers. What is most shocking is the lack of populations between these two peaks. The experimenter could artificially set up a generation with four purely discontinuous agents and one almost purely pre-verbal agent. And yet, iterated learning does not introduce such a generation: all five agents "grow up" in the same linguistic environment, and if this environment is such that four of them end up with a purely discontinuous grammar, then the fifth one will also have learnt a language as similar as possible.

To summarize, our SA-OT model of sentential negation cast in an iterated learning framework has not (yet) reproduced the entire story, but could mimic the S-shaped change from the pre-verbal stage to the discontinuous stage. Similarly, in a reversed experiment, the initial generation set to the post-verbal stage evolved into a population of discontinuous negation, via a mixed stage. Our model of Jespersen's Cycle can thus be compared to a pendulum: the two extreme positions, pre-verbal and post-verbal negation, are unstable, whereas the middle one, discontinuous negation, is a stable attractor. It is unclear yet why our "pendulum" would swing beyond the middle position. Maybe due to external factors, such as the phonological weakening of the SN morpheme. We have shown, however, that the S-shaped transition on a population level can be modeled even if, on an individual level, no grammar produces discontinuous negation between 50% and 100%.

7. Conclusions

The aim of this paper was to assess the validity of SA-OT as a model for linguistic change. In order to do so, we decided to look at the possible ways European languages express sentential negation and the way these strategies vary diachronically (Jespersen 1909, Jespersen 1917). We took as starting point the model developed by de Swart (2010) with its OT constraints. Our model, in section 5, was able to reproduce the three main stages of the evolution of sentential negation, corresponding to the types 1. pre-verbal, 2. discontinuous, and 3. post-verbal. It also reproduced the two mixed, transitional stages, and correctly predicted the lack of a third mixed stage between pure stages 3 and 1.

Although both de Swart's model and ours employ the same four constraints and consider the same six hierarchies, they make different predictions. De Swart's model predicted that the six hierarchies correspond to the three pure stages (see table below), and that the simple movement of one constraint triggers the transition from one stage to another. She also claimed that in principle the transition from stage 3 to stage 1 can be reproduced in the same way. In our model, however, we have shown, each hierarchy corresponds to a different stage (with the exception of Hierarchies 3 and 4), and there is no way to reproduce a direct transition between the post-verbal and the pre-verbal sentential negation types.

	Hierarchy	de Swart	SA-OT
1.	*Neg ≫ NegFirst ≫ NegLast	pre-verbal	pre-verbal
2.	NegFirst ≫ *Neg ≫ NegLast	pre-verbal	pre-V and discont.
3.	NegFirst ≫ NegLast ≫ *Neg	discontinuous	discontinuous
4.	NegLast ≫ NegFirst ≫ *Neg	discontinuous	discontinuous
5.	NegLast ≫ *Neg ≫ NegFirst	post-verbal	discont. and post-V
6.	*Neg ≫ NegLast ≫ NegFirst	post-verbal	post-verbal

More importantly, the models also differ in their methodology. De Swart is less concerned with the triggers of the change, not really elaborating on the reasons for two constraints being reranked. She contents herself with the observation that languages in a mixed stage correspond to a Stochastic OT grammar with two constraints being ranked very close, and thus getting frequently reversed. Hence, historic change is accounted for by a gradual change in constraint ranking, causing a gradual shift in the distribution of the produced forms.

Our model, however, tested explicitly the hypothesis that historic change is driven by imperfect mental computation ("performance errors") and imperfect learning (sections 5 and 6). The partial success of this novel enterprise shows that the question is far from being trivially soluble. Still, we hope that reconsidering some parameters may bring us closer to a fuller account of Jespersen's cycle.

Acknowledgment

The second author gratefully acknowledges the support of the *Netherlands Organisation for Scientific Research* (NWO, project number 275-89-004).

```
ALGORITHM: Simulated Annealing for Optimality Theory
 Parameters: w_init, K_max, K_min, K_step, t_max, t_min, t_step
 w := w_init ;
 for K = K_max to K_min step -K_step
     for t = t_max to t_min step -t_step
          Randomly select w' from the set Neighbors(w) ;
          C    := highest ranked constraint such that C(w) != C(w') ;
          k(C) := K-value of constraint C ;
          d    := C(w') - C(w) ;
          if ( d < 0  or  H(w) == H(w') )
              then  w := w' ;         # move to not-less harmonic neighbor
              else  w := w' with transition probability
                      P(C,d ; K,t) = 1         , if k(C) < K
                                   = exp(-d/t) , if k(C) = K
                                   = 0         , if k(C) > K ;
          end-if
     end-for
 end-for
 return w
```

Figure 8: The Simulated Annealing for Optimality Theory Algorithm (SA-OT).

Appendix: Pseudo-code and Parameters of the SA-OT Algorithm

The *Simulated Annealing for Optimality Theory Algorithm* (SA-OT; for introductions, see Bíró 2005, 2006 or 2009) is reproduced in Figure 8. Being a heuristic optimization algorithm, it models the imperfect computation performed by the human mind when it searches for the optimal element of the OT candidate set.

As discussed elsewhere (Biró 2007), the ranking values modified by the learning algorithms to determine the "highest ranked constraint such that `C(w) != C(w')`" is conceptually different from the *K-values* (the `k(C)` introduced in the next line of the pseudo-code) determining the transition probabilities. In the current experiments, however, the *K-values* of the constraints were chosen to be the same as their ranks.

The default ranks were: 4 for the highest ranked constraint Faith[Neg], and 3, 2 and 1 for the markedness constraints, in decreasing order following the hierarchy. The results presented on Figure 5 were obtained by diminishing the rank (and, hence, the K-value) of the lowest ranked constraint, NegLast, even further.

The starting point of the random walk, parameter `w_init`, was the candidate with a bare [V] as the surface form. A different strategy could have been to choose one of the infinitely many candidates that are faithful to the input, which contain a negation marker. Yet, this option is almost equivalent to let the random walker "walk away freely" from its starting point at the beginning of the simulation, before temperature drops to the range where the walk is influenced by the landscape. To test the effect of this initial phase, the standard parameter value `K_max` $= 5$ was once replaced by `K_max` $= 10$, but – unlike in Bíró

(2006, Chapt. 6) and (2009) – no significant change in the behavior of the model could be observed.

Parameter `K_step` was standardly set to 1. Instead of waiting for variable `K` to reach `K_min`, we introduced a counter that was increased each time the random walker did not move. The outer loop of the SA-OT Algorithm on Figure 8 stopped whenever the random walker had not moved for 50 consecutive iterations, because such a situation happens almost only if the random walker has reached a local optimum. Thereby we could avoid running the algorithm for too long or too short.

Finally, we used the standard parameter settings `t_max` = 3, `t_min` = 0, as well as `t_step` = 1. To our surprise, and differently from previous SA-OT models, tuning `t_step` did not significantly affect the behavior of the system. The most significant change was observed when the *difference* between the ranks of the two lowest constraints was increase, as discussed in section 5.

References

Bíró, Tamás (2005), When the hothead speaks: Simulated Annealing Optimality Theory for Dutch fast speech, *in* Cremers, Crit, Hilke Reckman, Michaela Poss, and Ton van der Wouden, editors, *Proceedings of the 15th Meeting of Computational Linguistics in the Netherlands (CLIN 2004)*, Leiden.

Bíró, Tamás (2006), *Finding the Right Words: Implementing Optimality Theory with Simulated Annealing*, PhD thesis, University of Groningen. ROA-896.

Biró, Tamás (2007), The benefits of errors: Learning an OT grammar with a structured candidate set, *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, Association for Computational Linguistics, Prague, Czech Republic, pp. 81–88. http://www.aclweb.org/anthology/W/W07/W07-0611.

Biró, Tamás (2009), Elephants and optimality again: SA-OT accounts for pronoun resolution in child language, *in* Plank, Barbara, Erik Tjong Kim Sang, and Tim Van de Cruys, editors, *Computational Linguistics in the Netherlands 2009*, LOT Occasional Series, LOT, Groningen, pp. 9–24.

Biró, Tamás (2010), OTKit: Tools for Optimality Theory. A software package. http://www.birot.hu/OTKit/.

Boersma, Paul (1997), How we learn variation, optionality and probability, *IFA Proceedings* **21**, pp. 43–58.

Boersma, Paul and Bruce Hayes (2001), Empirical tests of the Gradual Learning Algorithm, *Linguistic Inquiry* **32**, pp. 45–86. Also: ROA-348.

Chomsky, Noam (1965), *Aspects of the Theory of Syntax*, MIT Press, Cambridge.

Dahl, Östen (1979), Typology of sentence negation, *Linguistics* **17** (1–2), pp. 79–106.

Jespersen, Otto (1909), *Modern English Grammar on Historical Principles*, Vol. 1, Einar Munksgaard, Copenhagen.

Jespersen, Otto (1917), Negation in English and other languages, *Linguistica: Selected Papers in English, French and German*, 1933 ed., Munksgaard, Copenhagen.

Kirby, Simon and James Hurford (2002), The emergence of linguistic structure: An overview of the iterated learning model, *in* Cangelosi, Angelo and Domenico Parisi, editors, *Simulating the Evolution of Language*, Springer, New York, pp. 121–148.

Niyogi, Partha (2006), *The Computational Nature of Language Learning and Evolution*, MIT Press, Cambridge, MA – London, UK.

Pater, Joe (2008), Gradual learning and convergence, *Linguistic Inquiry* **39** (2), pp. 334–345.

Prince, Alan and Paul Smolensky (1993/2004), *Optimality Theory: Constraint Interaction in Generative Grammar*, Blackwell, Malden, MA. Originally published as *Technical Report nr. 2. of the Rutgers University Center for Cognitive Science* (RuCCS-TR-2).

Smolensky, Paul and Géraldine Legendre, editors (2006), *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*, MIT Press, Cambridge, MA – London.

Swart, Henriëtte de (2010), *Expression and Interpretation of Negation: An OT Typology*, Vol. 77 of *Studies in Natural Language and Linguistic Theory*, Springer, Dordrecht, etc., chapter 3: Markedness of Negation.