# A Computational Evaluation of Case-Assignment Algorithms

Miles Calabresi
Advisors: Bob Frank and Jim Wood

*Submitted to the faculty of the Department of Linguistics in partial fulfillment of the requirements for the degree of Bachelor of Arts*

Yale University
20 April, 2015

## Acknowledgments

With many, many thanks to my advisors, Bob Frank and Jim Wood. I would also like to thank Maria Piñango and Raffaella Zanuttini, as well as my friends and peers, Melinda, Laura, Maggie, Coco, Ally, Justin, and Andy for their plentiful feedback. Thanks as well to `http://www.latextemplates.com/cat/theses` and /cat/title-pages for the free LaTeX templates. Thanks to Anton Ingason for his quick replies and free sharing of the IcePaHC. I take responsibility for errors in this document.

"In theory, there is no difference between theory and practice; in practice, there is."
— Unknown

# Abstract

This project seeks to evaluate how successfully a new theory based on syntactic structure models the distribution of morphological case in Icelandic. This algorithm for case assignment will be tested against a traditional, grammatical-function-based theory on a large number of sentences from a corpus.

Morphological case is a noun's syntactic license to appear in its environment based on dependencies with other constituents, expressed as morphology on the noun. Traditionally, it has been held that case on a noun phrase corresponds to a grammatical function. This correspondence induces an algorithm for assigning case to a noun in a syntactic tree, based on its function. This account, however, has failed to account for the distribution of cases observed in Icelandic.

A new theory, based on the structural relations of heads rather than grammatical functions, has been devised to model the Icelandic irregularities while still correctly predicting the cross-linguistic data that appears to be function-based. The theory claims that case is assigned based on lexical properties of heads and syntactic relations among constituents. While its algorithm for assigning case has been well motivated in theory and has succeeded on isolated examples, it has not been widely studied on large quantities of data.

This new structure-based algorithm is operationalized as a computer program and compared to the function-based one. Each algorithm is applied to syntax trees from a tree bank of over a million words. Disregarding the cases listed in the tree bank, the program marks the nouns in the trees for case (according to the algorithm at hand) and compares its assignment against the attested case in the corpus. Along the way, it keeps track of how many nouns the given algorithm has marked correctly. The relative scores of each algorithm will answer the question of how successful as a theory of case distribution the structural theory is compared to the traditional account.

# Contents

# 1 Introduction: Evaluation of a Structure-Based Theory of Case

The primary aim of this project is to evaluate two theories of case by modelling their methods on large quantities of data. The evaluation will consist of testing a case-assignment algorithm specified by a new theory of case (described in Section 3) on a tree bank of Icelandic and comparing the result to that of a more traditional theory of case (outlined in Section 2.2).

# 2 Background: Case in Icelandic

## 2.1 Working Assumptions about Case

I will take case, or more precisely, *morphological* or *surface case*, to refer to the systematic morphology that reflects dependencies between noun projections and other categories in a given utterance. Each case is the label given to the consistent morphological markers associated with a language-specific class of morpho-phonological and syntactic contexts. These labels are used only by convention, and I assume no intrinsic properties attached to any given case that might affect how it is assigned. That is, "nominative" is just a name given to the case that appears as morphology $X$ and is assigned in circumstances $Y$ and could just as easily be called "accusative." For this project, I consider only the four cases of Icelandic: nominative (abbreviated NOM), accusative (ACC), dative (DAT), and genitive (GEN).

All proposals discussed here assign case to nouns post-syntactically, a fact justified in McFadden (2004). Every theory considered therefore begins with a syntax tree, all of whose constituents lie in their surface positions. There is one exception that traces of moved constituents are used to determine the base position of nodes that have undergone A′-movement because case is assigned to the head of an A-chain. The theory then attempts to assign case to the noun heads in that tree (see Section 6.2 for discussion of why heads and not higher projections).

Furthermore, all of these theories claim to model only the distribution of the case assignment in the sense that they do not necessarily seek to represent the actions of the mechanism(s) that assign case inside the brain in real time. Rather, they model the abstracted process of assignment and the results it produces.

## 2.2 Traditional Theory: case assignment based on grammatical functions

It has been widely held in the past that the case assigned to a noun phrase corresponds to a grammatical function, such as subject, as given in (1). This view is prevalent in many grammars of Latin and other case-marking languages. This traditional theory, which I will refer to as the "Grammatical-Function–Based Algorithm" (GFBA) assigns case based on grammatical functions of noun phrases as follows:

(1) a. NOM → subject

b. ACC → direct object

c. DAT → indirect object (including object of a preposition)

d. GEN → possessor

Historically, this algorithm has appeared to explain case to a satisfactory degree and has been taken as the standard that so-called "exceptions" violate. See Butt (2006) for discussion. For instance, it correctly predicts the distribution of *I* (NOM) and *me* (ACC) in (most "standard" dialects of) English, as in (2), and in the Icelandic example (3a).

(2) a. I/*me went to the park.

b. John hit me/*I.

## 2.3 Case in Icelandic (Why Icelandic?)

The distribution of case in Icelandic has been notoriously difficult to reconcile with the Grammatical-Function–Based Algorithm. This traditional account fails to explain many phenomena, such as the *oblique* subjects, where subjects of some verbs systematically do not bear the expected nominative case. Some sentences, such as (3a), follow the traditional pattern. However, many sentences of Icelandic do not, for instance, (3b) and (3c).

(3) a. Trúð-urinn sendi Jón-i hest-∅ mann-s-in-s
clown.NOM sent John.DAT horse.ACC man.GEN-the.GEN
'A clown sent John the man's horse.' [1]

b. Harald-ur mun skila Jón-i pening-un-um í kvöld
Harold.NOM will return John.DAT money.the.DAT/*ACC tonight
'Harold will return the money to John tonight.'[2]

c. Mér sárnaði þessi framkom-a han-s
me.DAT/*NOM hurt this behavior.NOM/*ACC/*DAT he.GEN
'I was hurt [offended] by this behavior of his.'[3]

In sentence (3b), the direct object *peningunum* has dative case instead of the expected accusative. In sentence (3c), the subject[4] *mér* has dative case instead of nominative, while the object *framkoma* unexpectedly bears nominative. Thus, the traditional model of case-marking fails to explain the data of Icelandic.

Since it exhibits an "unusual" distribution of cases, Icelandic is a perfect ground to test theories of case: any such theory should be able to accommodate the Icelandic data.

# 3 The New Theory: A Structure-Based Algorithm

## 3.1 Overview of the Theory

McFadden (2004), Marantz (2000), and Wood (2011) have argued for a new theory of case. Their work is related to, though distinct from, ideas put forth by Yip, Maling, and Jackendoff (1987) and Sigurðsson (2012). The primary claim is that case is assigned based on structural relations within the sentence rather than grammatical functions. The

[1] This example is due to Jim Wood.
[2] Wood (2015, p. 134)
[3] Maling and Jónsson (1995, p. 75)
[4] Zaenen et al. (1985) show that *mér* is a real subject (usually defined as the specifier of TP) not a fronted object as in Spanish *gustar* constructions. This fact is crucial to asserting that (3c) truly deviates from the pattern of a subject corresponding to nominative case.

portion of this theory that the current project seeks to evaluate is the algorithm for case assignment, which I will refer to as the Structure-Based Algorithm (SBA).

The algorithm models the distribution of case in a sentence by checking four conditions. If each condition, in order, applies to any unmarked noun (a noun which has not yet been marked with a case) in the sentence, the algorithm assigns case to that noun as specified by the condition at hand. The order in which the nouns are tested for each condition does not matter. This check-and-assign process is repeated until the condition doesn't apply to any unmarked nouns in the tree. The algorithm then moves on to the next condition, stopping when all nouns have been marked with a case. The last condition is a "default," so exactly one condition should always apply to a given noun.

## 3.2 Steps of the Algorithm

The specific steps of the algorithm are as follows. At any given step, the nouns in the sentence may be considered in any order because each step's conditions will specify at most one case for any given noun in the sentence.

(4) A. Lexically Governed Case: certain heads listed in the lexicon license case, which is also listed in the lexicon, on one or more of their arguments

B. Dependent Case: unmarked nouns are assigned case based on structural relations between them

C. Unmarked Case: unmarked nouns are assigned case based on their local environments

D. Default Case: all remaining unmarked nouns are assigned one (language-dependent) case

Next, we discuss each step with a brief motivation. The specific implementations are delayed until Section 7, where necessary approximations are discussed in detail.

Suppose the SBA is given a syntax tree, complete with category labels. Then the algorithm executes the above steps as follows.

### Step 1

The *lexical step* consists of two parts. The first is somewhat of a "black box." It supposes that certain items (here, verbs and prepositions) just assign case anomalously to their arguments. This substep searches through each node of the tree for lexically specified ("quirky") verbs and prepositions. Some or all of each such verb's (or preposition's) arguments are then assigned case according to the verb's *case frame*, information which is located in the lexicon. The algorithm searches for the appropriate argument(s) (subject, direct object, and/or indirect object for verbs, and the object of a preposition) and marks them as specified by the case frame. The order in which the arguments are sought does not matter because any given noun will not be an argument of more than one verb (or multiple arguments of the same verb). Since this part of the lexical step is a theoretical concession to some measure of irregularity, it will also tested as the first step of the GFBA.

The second part of the lexical step assumes that *applicative* heads (a kind of *v* head that, among other things, introduces experiencer arguments of the main verb) license the assignment of dative case to their DP specifiers and that (non-quirky) prepositions do the same for their DP complements. This part of the step assigns case only to unmarked nouns in the tree. This means that if a noun was marked with a case in the previous substep, one does not consider it here.

### Step 2

The *dependent step* continues with the nouns that are unmarked by this stage. For each unmarked noun (again, considered in any order), this step checks whether any of the other unmarked nouns in its *minimal domain* c-command it. A node in a tree is said to *c-command* another if neither dominates the other and all branching nodes that dominate the first node also dominate the second. McFadden (2004) defines a *domain* as a phase ($v$P or CP), though I adjust this choice of categories in Section 6.4. A domain containing a given node or set of nodes is called *minimal* if any other domain that contains the node(s) also contains the first domain.

Given these definitions, we return to the algorithm. If such a configuration of an unmarked DP c-commanding another unmarked DP in a minimal domain is found, then the lower noun (the one that is c-commanded) is assigned accusative case.

A noun may c-command and thus license accusative on more than one other noun in this step. This fact makes this step appear to be performed from the bottom up in the sense that if there is a chain of three or more nouns which c-command one another, then both/all of the lower ones are assigned accusative. However, due to the restriction that all nouns being considered must lie in the same domain, it does not make a difference in what order the case assignments are performed on the lower nouns because the highest noun in the domain will always c-command all of the lower ones and will never be marked with dependent accusative itself (which one might think *a priori* could block its licensing case on lower nouns). Note that this step resembles the assignment of accusative case to direct objects in the GFBA because subjects often c-command their verb's objects in a minimal domain.

#### Step 3

The *unmarked step*, too, begins by considering all thus-far-unmarked nouns. This step assigns case to nouns based on their local (minimal) environments. The *environment* of a given noun is the first ancestor of the noun's maximal projection (so as not to count projections the noun itself; see the discussion of NP-approximation in Sections 6.1 and 6.2) that is either a CP, an NP, or a PP. Inside each minimal environment (in the sense of minimal given above), a different case is assigned to all unmarked nouns lying in it. If a noun's minimal environment is a CP, then it is assigned nominative; if it is a NP, genitive; PP, dative. This process is continued until there are no unmarked nouns that lie in a domain. Since a noun's minimal environment does not depend on how (or whether) other nouns are marked, the order in which nouns are considered does not matter in this step either. Though these case-environment pairings (nominative with CP, and so on) are common, the theory allows them to vary across languages.

Once again, the assignments made in this step roughly correlate with the function-based theory inasmuch as subjects tend to lie in clauses, possessors are possessed by some other noun that (locally) dominates them, and objects inside a prepositional phrase are usually indirect objects of a sort. However, these tendencies are nothing more: for instance, subjects in embedded clauses that have only a TP and no CP layer would not necessarily be assigned nominative in this step. While this step will, in general, leave very few nouns unmarked because most nouns are (eventually) dominated by a CP, the following step is important for assigning case to any nouns that do not.

### Step 4

The final step is the *default*. Continuing again with the remaining unmarked nouns in the tree, it assigns all such nouns a default case, which varies from language to language. For Icelandic, the default is nominative. In English, it is accusative. This default step ensures that all nouns are marked at some point, and as before, the order in which nouns are assigned the default does not matter. As mentioned above, the default step will not apply frequently.

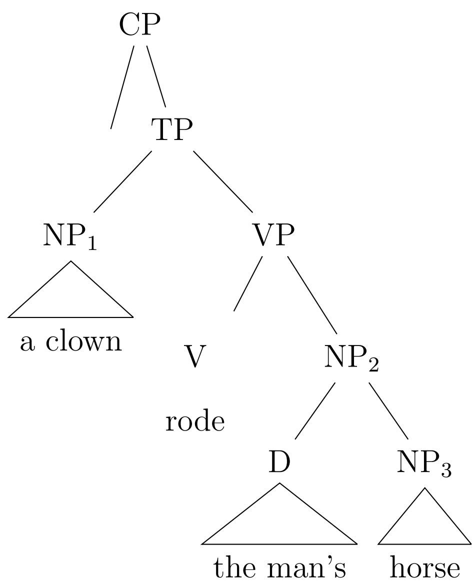
### A Worked Example

To conclude this section, the algorithm is applied an example to demonstrate how the steps work. The example tree's structure is simplified compared to what the theory assumes (exact details are not required to illustrate the algorithm at work).

(5) a clown.NOM rode the man.GEN horse.ACC

1. *ride* is not a quirky verb (I decree so for this example); do nothing
2. since *a clown* c-commands *the man's horse* in a minimal CP, mark *horse* ACC; *the man* is not in the same minimal domain as *a clown* (and is therefore not marked in this step) since it lies within the $NP_2$ node headed by *horse*
3. within the environment $NP_2$, mark *man* GEN; within the matrix CP, mark *clown* NOM.

   All nouns are marked, so we stop. No need to use step 4.




### 3.3 A Note on this Algorithm

The SBA as presented above is my synthesis: it is my interpretation of information from multiple sources. All results in the following tests are based on the algorithm as described here, and any errors and misinterpretations in the algorithm are my responsibility.

## 4 Questions and Hypotheses

The SBA appears to work well on individual examples in McFadden (2004), but the aim of this project is to examine its claims on a large number of sentences from "real world"

Icelandic documents. The large-scale data is taken to be a proxy for the whole language, and therefore we try to draw conclusions about the viability of the SBA as a model of case in Icelandic (and by extension in other languages).

To be clear, this evaluation provides only one perspective. Since it only looks at the data and ignores the theoretical underpinnings of the algorithms, it is a necessary but not sufficient condition for a "veritable" Theory of Case. For instance, it turns out that quirky verbs' arguments account for a very small percentage of nouns, so it would be possible for an algorithm to mark nearly all nouns correctly while ignoring quirky verbs entirely. Such an algorithm would score well on a data-driven evaluation but might not be considered "good" because it ignores a well known, if not frequent, aspect of case theory.

The primary goal of this project, then, is to answer the question of how well the algorithm as described above works on big data. Specifically, is it correct more often than the GFBA? How well does each algorithm do absolutely? The answers will come via a computational evaluation: mechanizing the algorithm and testing it on a corpus of Icelandic. The number of correct assignments that it makes out of the total number of nouns in the corpus will be used as the most direct evaluation of the algorithm.

The main hypotheses is that the SBA will perform better than GFBA, which will perform better than the baselines.

# 5 Method

Thus far, my uses of "how well" or "how often" have not been specific when it comes to evaluating each algorithm (the SBA and GFBA). This section describes the tools used to implement these algorithms and the procedure to evaluate them, both "in absolute" and relative to each other. Limitations and difficulties associated with using these materials are discussed later in Section 6.

## 5.1 Materials

### 5.1.1 The IcePaHC

As stated above, the primary aim of this project is to evaluate the SBA on large quantities of "real world" Icelandic data. Wallenberg et al. (2011) provide that data in the form of the Icelandic Parsed Historical Corpus (the IcePaHC). The IcePaHC contains over one million words in sixty-one documents from 12$^{\text{th}}$ to 21$^{\text{st}}$ centuries, but the algorithms are tested on the four documents from the 20$^{\text{th}}$ and 21$^{\text{st}}$ centuries. The number and diversity of these sources is discussed in Section 9.5. In the IcePaHC, noun heads are marked with case, which is used here as the standard of correctness.

### 5.1.2 Lexical information

The second source of data I make use of in evaluating the SBA is a list (included in Appendix B) of the behavior of prepositions and "quirky" verbs in Icelandic. I compiled this information from Barðdal (2011), Jónsson (2000), Jónsson (2003), Jónsson (2009), Tebbutt (1995), and conversations with Jim Wood to drive the lexical step. The ways in which the information from these sources was combined are discussed in Section 6.5.

### 5.1.3 A program to test algorithms

To tie the whole experiment together, I have written a Python program to mark case on IcePaHC trees according to rules specified by the GFBA and SBA. The program reports the results of the case marking in terms of the number of nouns that are correct (in agreement with the case as marked in the corpus), as well as a few other statistics. A link to the program's source code is provided in Appendix A, and I describe its implementation of the two algorithms in Section 7.

## 5.2 Procedure

The procedure consists of running the program, configured to implement a given algorithm with a given set of lexical information, on a batch of documents from the IcePaHC. Each such run of the program with a different configuration of these parameters will be called a trial. The program will score each trial according to the number of correctly marked nouns and compare scores across algorithms.

### 5.2.1 Algorithms to be Tested

- Structure-Based Algorithm (described in Sections 3.2 and 7)
- Grammatical-Function–Based Algorithm (described in Section 2.2, but with lexical step described as part of the SBA)
- Three baseline algorithms, each a different level of “non-theory”
  - truly random marking of each case (each noun has a 25% chance of getting each case)
  - random marking in proportion to the frequency of the cases in the document(s) at hand
  - uniform marking of the most frequent case (nominative)

### 5.2.2 Scoring

To put the algorithms’ scores in context, they will be compared to one another and to the baselines’ scores. The goal of doing the latter is to give a sense of absolute success: any algorithm must beat the baselines in order for the theory to be considered viable. For each trial, in addition to the raw number of correct case assignments, the program calculates three measures for each case: precision, recall, and f-score.

For a given case X, *precision* is the proportion of nouns the algorithm marked correctly as case X out of all nouns it marked as X. *Recall* is the proportion of nouns marked correctly as case X out of all instances of case X in the corpus' annotations. Precision rewards careful marking and penalizes catch-all tactics (such as guessing nominative when one is unsure because nominative is most frequent), while recall rewards broader coverage and penalizes cautiousness. The *f-score* combines precision and recall into a single number between zero and one, $2\frac{p \cdot r}{p+r}$.

F-score is a useful measure because it combines the other two in such a way that a good score by one measure will not compensate for a bad score on the other. It is easy to maximize either precision or recall with simple heuristics, but doing so will produce a very low score on the other. The f-score balances out these extremes by giving a mediocre score, while rewarding algorithms that score well on both. Furthermore, since the f-score is a harmonic mean of two ratios, it does not give higher weight to a higher number. That is, a particularly high (or low) score on one measure will be given equal weight as a moderately high (low) score, which makes sense when considering two ratios.

The average of the four f-scores for each case is also presented. In a sense, that average f-score measures how well the algorithm performs in theory (not just in practice) because in order to score well on an average of the four cases that is unweighted by frequency, an algorithm must score well on all four cases individually. For example, an algorithm could, in principle, ignore genitive case entirely (which occurs on approximately 7% of all nouns in the IcePaHC) and still score 93% correct. However, the zero f-score for genitive would drag down the average f-score to (at best) a 75%, far more than genitive's 7% weight, and thus "rightfully" penalize the algorithm for disregarding an important aspect of the theory.

The raw scores and average f-scores for each of the algorithms are used to compare them, both against the baselines and against one another. Again, the ultimate goal is to use the relative scores of each trial to answer the question of how successful the Structure-Based Algorithm is as a theory of case assignment.

# 6 Approximations and Limitations in Materials

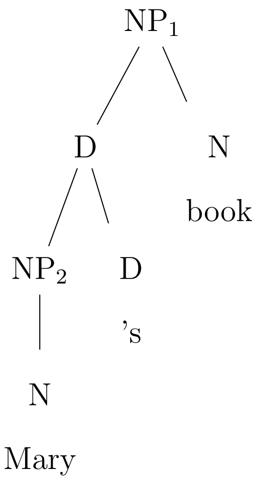
There are a number of approximations I made in implementing the program to work with the lexical information and the conventions of the tree bank. While it is difficult to list all of the individual choices needed to interface the IcePaHC with the theory of the SBA, several major ones are described here.

## 6.1 No DPs in the IcePaHC

In the IcePaHC, NPs are the highest projection of a noun – there are no DP layers. This structural assumption introduces some important practical distinctions when it comes to implementing the algorithm. Specifically, there are two tasks that do not work perfectly: finding the head of a given NP layer and determining whether an N head is part of a given NP that serves a given grammatical function.

The crux of both issues is the question of whether a given NP layer is a projection of a given N head. For instance, since example (6) has the NP as the highest layer, it is difficult for the program to tell whether *Mary* or *book* is the head noun of the phrase *Mary's book* without more information.

(6) Mary's book




In particular, it is not clear how to determine that at step 3 of the SBA, *Mary* should get genitive from being inside $NP_1$ but not $NP_2$, while *book* should not get genitive from being inside $NP_1$.

To answer this specific question, I use POS tags for tagging genitive possessors. That is, NP-POS is the unmarked environment for genitive rather than just NP.

For the general problem of finding the maximal projection of a noun head, I used information about what layers commonly intervene between N nodes and NP nodes. There are eight common categories that come between N heads and NP layers in the tree bank's modern documents: IP, CP, PP, NP (including WH noun phrases), NX, QP (for pronominal Q heads), CONJP, and non-structural CODE annotation layers. I choose to treat the first three as boundaries but look past the last five as possible intermediate layers of the maximal projection headed by the N node at hand. Starting from each unmarked noun head, the program looks up at ancestors, ignoring intervening CODE, (WH)NP, NX, QP, and CONJP nodes. Once it hits a node that is not one of those five, it stops, assuming it has found the maximal projection of the head (unless the last node is a CONJP or CODE, in which case the program backtracks down one generation).

For the converse problem of finding heads for the NP-SBJ nodes that the lexical step locates, I use the heuristic of searching the NP's children for an N head. If there are none, search for NP children that are not NP-POS nodes (as the head of an NP will not be the possessor of the same NP) and repeat this process on the leftmost non-possessor NP child.

## 6.2 Case is assigned to noun heads

While McFadden (2004) argues that case is assigned to determiner phrases, from which it percolates down to the noun (and possibly other) heads, where the morphology is realized, the IcePaHC annotates case on the N heads themselves. As such, I mark case on the noun heads to facilitate comparison with the corpus.

In implementing the algorithms, I use the above functions to find an N head's maximal projection and to find an NP's head heavily. This capacity to switch between the two enables the program compare the maximal (would-be DP) nodes for things like residing in the same domain or standing in a c-command relation while still assigning case to the heads so they can be easily compared to their case-marked counterparts in the IcePaHC.

## 6.3 No applicative heads in the IcePaHC.

In the SBA as advanced by McFadden (2004), dative case is assigned to (some) indirect objects by applicatives (described above in Section 3.2). However, the IcePaHC does not use applicatives. Therefore, I have made the following approximation to model the assignment of dative as closely as possible. I use the NP-OB2 and rare -OB3 tags to identify indirect objects, despite the fact that these tags are functional. The applicatives that McFadden assumes to assign dative introduce indirect objects, so it is not against the structural spirit of the algorithm to use this function-based information: it is approximating the assumptions McFadden makes about the structure surrounding indirect objects.

## 6.4 Case domains not completely specified

McFadden argues that the domain for Step 2 of the SBA is a phase (CP or *v*P). Since the IcePaHC contains no *v*Ps, it is tempting to use IP as an approximation. Unfortunately, that is not accurate enough: for example, IPs would incorrectly block ECM, and certain subclasses of IPs (such as small clauses and participial clauses) should never be considered boundaries. I therefore use just CP as the domain boundary. Since this choice of just CP does not block the assignment of dependent accusative to NP-internal possessors (as a *v*P should in most sentences), the assignment of genitive in the unmarked environment of NP-POS is relegated to the second half of the lexical step, as described below in Section 7.

## 6.5 Redundant, Expected, and Conflicting Quirky Case Frames

Across and within some of the consulted sources, there are some redundant, expected, and apparently inconsistent case frames (the paradigms for which case is assigned to which arguments of the quirky verb) reported for the same verb. Redundant frames appear in two varieties: the same frame reported from multiple sources, or two frames that are consistent but one is more specific (such as one source specifying a dative subject while another specifies a dative subject and a dative direct object). I simply merged frames into one, selecting the frame that specifies more arguments over the one that specifies fewer.

Other times some of the lexical information specified is exactly what one would expect the SBA (and GFBA, for that matter) to produce if the given item (verb or preposition) were not treated as an exceptional lexical item, or if only part of its listed behavior were executed in Step 1. For this reason, trials will be run without the following "expected" case frames:

- nominative subjects
- accusative direct objects when the subject is *not* lexically specified
- dative indirect objects [specified in Step 1b]

Though certain configurations, such as nominative objects when the verb is lexically specified, are expected, they never occur in the list of quirky verbs' case frames, and therefore I do not explicitly exclude them here. Indeed, Wood and Sigurðsson (2014) argue that there are no predicates lexically specified to take a nominative object.

Finally, there were some verbs that were presented with multiple different case frames. Sometimes both frames may be possible but induce a semantic difference by changing the meaning of the verb or preposition and its argument(s). There is also known to be cross-speaker variation in the case frames of some quirky verbs. The solution to this problem is to try all variations and see if any is correct. This strategy is related to the following subsection, where I justify not marking quirky case if it is not the same as the case given in the corpus.

## 6.6 Advantage to the first part of the Lexical Step.

Though this is not specified explicitly as part of the lexical step, I have implemented the lexical step in such a way that it is never wrong (by the standards of the IcePaHC). The motivation is that multiple case frames and variation in the behavior of quirky verbs provide for tricky assignment of lexical case.

The justification for this modification is that many verbs alternate their quirky assignments with other case assignments (whether expected or a different quirky case frame), so it is unfair to penalize the theory for instances where the verbs fail to mark their arguments with the quirky cases that the algorithm happens to know about. The algorithm (which is meant to allow for cross-speaker variation) is trying to generate the case patterns seen in the corpus, but its lexical information might differ slightly from that of the speaker who generated the given tree. Therefore, the program assumes that if the algorithm's lexical information predicts the case observed in the tree bank, then the lexical information matches the speaker's, and the assignment is made. On the other hand, if the cases are not the same, the program assumes the lexical information does not match the observed case was assigned by a different process.

As an analogy, consider a program attempting to evaluate a text-to-speech algorithm for realizing strings of letters as phonemes. The algorithm might know that "e" is mapped to /$\varepsilon$/ as a general rule. However, one might tell the algorithm that (for some speakers) "e" should be realized as /a/ in the context of "_nvelope". The algorithm should not be penalized for knowing to try /a/ if the initial "e" in "envelope" happens to be realized as /$\varepsilon$/ in a particular instance in the corpus. Likewise, the SBA is not penalized if the lexically-specified case of a quirky item does not match the case of that item in a given sentence of the corpus.

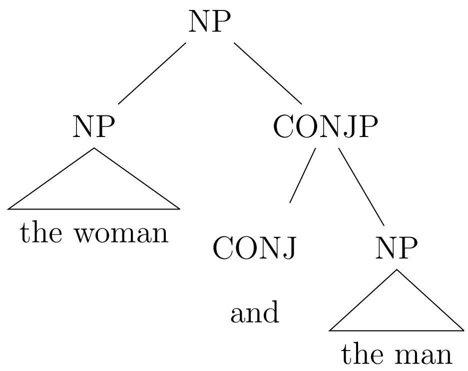
Furthermore, the lexical step is already something of a "freebie" – it's supposed to describe the behavior of anomalies, so it says little about the theory (though perhaps something about the limited lexical information I use) if its assignment not match the instance at hand.

For these reasons, I implement the lexical step in a way that it never marks a case that disagrees with the one found in the IcePaHC. It will mark the case specified in the "lexicon" for a given noun argument of a given quirky verb (or preposition) if and only if it agrees with the actual case in the corpus. Otherwise, it leaves the noun unmarked. Note that this advantage is not given to the second part (where dative indirect objects and genitive possessors are assigned).

## 6.7 Conjoined NPs not ignored

As described at the beginning of Section 7, all conjoined noun heads but the first are ignored so as to avoid double- (or triple-, etc.) counts of what is essentially the same case assignment. However, some nouns are joined at levels above the head, as in example (7).

(7) Conjoined NPs




It is an unfortunate, if not frequent, error in the numbers that the case assigned to such nouns will be counted multiple times.

## 6.8 Null arguments not counted as nouns

In the IcePaHC documentation, case is marked on non-nominative empty subjects. For simplicity, and so as not to skew the number of subjects, the program does not count such subjects as nouns. Other null arguments do not have case marked on them and therefore are not counted as nouns either. These would choices affect Step 2 of the SBA because these null arguments are not counted as unmarked nouns, so they don't license accusative on lower nouns in their domain. In order to counteract this scenario in part, the program does count null subjects as possible licensers of dependent accusative in Step 2 of the SBA.

## 6.9 Quantifiers not treated like pronouns

Though I treat pronominal quantifiers (which I distinguish from modifiers by counting only Q heads in the IcePaHC that do not have N siblings) as nouns, the IcePaHC does not treat them as it does (pro)nouns. They are not given function tags, and they don't always project phrases. In these situations, they will not be marked by the GFBA (the effect of this inconsistency on the GFBA's results is discussed in Section 9.4).

## 6.10 Multi-word quirky predicates approximated

Given the form of the tree bank, it is difficult to locate particles and other words that may form multi-word quirky predicates, such as the adjective *kalt* in the quirky predicate *verða kalt* ("get cold"). This adjective may be moved or simply not occur right next to the verb *verða*. There are several other constructions that complicate the matter further.

Given the advantage to this part of the lexical step discussed in Section 6.6 (that it will never mark a quirky case that disagrees with the case listed in the corpus), I make the following approximation for simplicity's sake. When searching for multi-word predicates in the tree bank, I test only the first word, which is almost always the verb. Therefore, the algorithm will attempt to mark the arguments of any instance of the verb *verða* according to the case frame associated with the lexical entry *verða kalt*. If the case frame is agrees with the case listed in the tree bank, then the assignment is made. If not, then no assignment happens (though since there are several *verða* predicates, the algorithm may attempt to give same argument case multiple times with different case frames). This approximation is helped by the fact that verbs that combine with predicate adjectives or nouns often have the same case frames, decreasing the likelihood that a correct assignment is made by chance.

### 6.11 IcePaHC trees are very flat

While there are several problems that arise from the discrepancies between the IcePaHC's flat trees and the sort of structures assumed in McFadden (2004), one of the biggest is the over-assignment of dependent accusative in the Step 2 of the SBA. That step says that any unmarked noun that c-commands another unmarked noun (in a minimal domain) licenses dependent accusative on the lower noun. However, due to the flat tree structure, many nouns that would lie at different levels within the tree according to a McFadden-style structure symmetrically c-command each other in the IcePaHC. This causes a huge over-assignment of dependent accusative where it is not intended to happen. In order to compensate, I add linear precedence as a condition to license dependent accusative. That is, a noun must c-command and precede another in order to license accusative.

## 7 Implementation

In the previous section, we saw several challenges that arise in reconciling the various assumptions of the IcePaHC and lexical information with the algorithm rules. In this section, I describe exactly how I implemented the GFBA and SBA to run on the IcePaHC, emphasizing the necessary approximations and possible errors that this implementation introduces.

## 7.1 General Implementation

Three kinds of nouns are excluded from being eligible to receive case. Appositives, all nouns but the first in a coordinated series (joined by conjunctions), and all proper nouns but the leftmost string of siblings. Thus, in the following example, *rust bucket*, *Smith*, and *Boris* are not counted as nouns by the program even though they are all labeled as distinct nouns in their own right in the tree bank.[5]

(8) Excluded nouns

a This car, a real *rust bucket*, won't get you there.

b Mary *Smith* traveled to Japan.

c Amy and *Boris* played basketball last night.

I assume that all three of these classes are assigned the same case as their immediate predecessor. While I could not find it mentioned explicitly in any source I consulted, McFadden (2004) implies that at least appositives and conjuncts should bear the same case as the nouns they modify or are coordinated with. The GFBA, as an empirical observation about case, would not obviously predict a different assignment for these three classes, either. Therefore, since the theories do not differ in their predictions, these classes of nouns are excluded in order not to double-count the case assignments.

The program begins by taking all of the IcePaHC trees as input and performing the following operations on them one at a time. Given a tree, it creates a copy and removes all case markings from noun heads, using a placeholder symbol. It then marks all of the nouns in each tree for case according to the given algorithm for the trial. If no rule applies to a noun, it is left unmarked (with the placeholder symbol).

[5]Last names such as *Smith* are given a separate projection in the IcePaHC because last names may bear a different gender than first names in cases of patronymy or conjoined names such as *John and Jane Doe.*

As discussed in Section 6.2, case is assigned to head nouns in all algorithms because the IcePaHC marks it there (rather than at the NP or DP level).

Whenever the algorithm is searching for a maximal projection or a head, it follows pointers of A′-movement along the way so that if it reaches a constituent that is A′-moved, it looks to the base position and uses the immediate surroundings there as if they were the surroundings of the moved constituent.

## 7.2 Quirky Assignments (Lexical Step)

The program begins the lexical step by searching through the tree for verbs and prepositions. For each such node it finds, it checks it against a list of items that lexically govern the case of some or all of their arguments. This list is a stand-in for a corner of the lexicon. It is described in more detail in Section 5.1.2 and is reproduced in Appendix B.

The program then attempts to locate the argument(s) that the verb (or preposition) lexically specifies and assign them case according to the verb's case frame (the specified case for each argument). When looking for these arguments, the program uses a CP boundary (the same as a case domain from Step 2) combined with the conditions that subjects c-command the verb while objects be c-commanded by the verb, and that the argument have the appropriate grammatical function tag. As described in Section 6.6, the algorithm assigns case to the argument if and only if the case specified for the given argument by the case frame agrees with that argument's case as given in the tree bank.

## 7.3 GFBA

First, the quirky portion of the lexical step, as described above, is run. Entering the non-lexically-specified part of the algorithm, the program begins by looking for unmarked noun heads. The program considers the noun heads in any order it finds them because the order in which they are assigned case doesn't mater. (I have tested it in all orders and the results don't change!) It then searches for grammatical function tags – subject (SBJ), direct object (OB1), indirect object (OB2 and rare benefactive OB3), and possessor (POS) – on NP projections of those heads by first finding the noun's maximal projection following the scheme described in Section 6.1. It tests to see whether the maximal projection is one of NP-SBJ, NP-OB1, NP-OB2, NP-OB3, or NP-POS, and it checks whether the parent of the maximal projection is a PP (in order to classify objects of a preposition that were not captured by the lexical step). If the maximal projection is not marked as one of the grammatical functions and is not the daughter of a PP, then it is left unmarked. In this case, the GFBA has nothing to say about the noun.

For this algorithm, the above challenge of determining whether a given N and a given NP are part of the same projection is not too difficult because a noun's function is closer to the head than any other environment.

## 7.4 SBA

### Step 1

The first half of the lexical step is the quirky assignments, as described above.

For the second half of the lexical step, the program first compiles a list of all unmarked noun heads and their maximal projections (in any order). If any unmarked noun on the list is an indirect object (determined as in the GFBA's implementation), it is assigned dative case. This step is an approximation for the applicative head McFadden (2004) assumes but which is not present in the IcePaHC (see Section 6.3). The same is done for objects of a preposition, again using the same procedure as the GFBA.

Finally, any heads of possessors (once again, these are located using the corpus' function tags, as in the GFBA) are marked genitive. This assignment is meant to take place in Step 3 when DPs whose local environment is another DP are assigned genitive. However, given the approximations in Section 6.1, this step is moved up here to avoid having to add NP-POS as a boundary to block dependent accusative in Step 2 as well as adding

NP-POS as the environment for genitive in Step 3. Whenever a noun is assigned case in this step, it is removed from the list of unmarked nouns.

#### Step 2

The program iterates over the list of unmarked nouns and compares the maximal projections of distinct pairs of nouns. If one maximal projection both precedes and c-commands another within a minimal domain (as defined in Section 3.2), then the other noun is assigned accusative. As before, each time such an assignment is made, the noun is removed from the list.

#### Step 3

The program continues using the list of unmarked nouns from the previous steps. It iterates over the list and, starting at each noun head's maximal projection, looks up through its ancestors until it reaches a CP, IP-MAT (matrix IP), PP, or NP-POS. When it finds one of these, the program assigns the noun nominative if it found a CP (or IP-MAT since matrix clauses are not all assumed to have a CP layer in the IcePaHC), dative for PP, and genitive for NP-POS. It removes any nouns it marks from the list of unmarked nouns. For the reason NP-POS is used instead of NP, see Section 6.1.

#### Step 4

Again iterating over the remaining unmarked nouns, the program assigns nominative to all nouns remaining on the list.

## 7.5 Scoring

As it executes an algorithm's assignments, the program compares the assignments it makes against the case given in the corpus, and it keeps track of how many nouns the given algorithm marks correctly. The total number of correctly marked nouns (in all trees) divided by the total number of nouns is the raw score for the algorithm. More refined scores are generated by various analyses described in Section 5.2.2. The scores for each algorithm allow the algorithms to be compared to one another, as well as to the baselines described in Section 5.2.1.

## 7.6 A Note on Labels

In the implementation of the program, I search for many tags (such as NP-) by matching prefixes or suffixes of nodes' labels. For instance, in looking for a noun phrase, the program asks whether the first two letters of a given node's label are 'N' and 'P' in that order. This technique is justified by a corpus count of the number of (for instance) occurrences of the string 'NP' as compared with the number of occurrences of the string '(NP'. If the two numbers are equal, then I assume that all noun phrases have labels that begin with the prefix 'NP' (because the open parenthesis indicates the beginning of a label). That is, these counts verify that all labels that contain the string 'NP' indeed begin with it. Therefore it is safe to search for NPs by looking only at the prefixes of labels. Similar counts justify the use of prefixes and suffixes for other categories.

Noun heads (and a few other categories, such as quantifiers) have many annotations, and for these categories, a more complex regular expression based on the IcePaHC documentation is used to identify them.

# 8 Results

All results below are from one of the algorithms as described above on the four modern texts from the 20$^{\text{th}}$ and 21$^{\text{th}}$ centuries. First, I present the actual distribution of cases in the corpus then the results from running each algorithm. Each algorithm's confusion matrix is presented, along with the precision, recall, and f-score by case. Finally, the total number of correct assignments, incorrect assignments, and nouns left unmarked are presented.

## 8.1 Frequencies of Cases in the IcePaHC

| Case | Count | Frequency |
|---|---|---|
| Nominative | 9,200 | 38.26% |
| Accusative | 6,369 | 26.49% |
| Dative | 6,683 | 27.80% |
| Genitive | 1,792 | 7.45% |

## 8.2 Baseline (all nominative)

Baseline Confusion Matrix

| Case assigned by algorithm → / Case listed in corpus ↓ | N | A | D | G | Unmarked |
|---|---|---|---|---|---|
| N | 9,200 | 0 | 0 | 0 | 0 |
| A | 6,369 | 0 | 0 | 0 | 0 |
| D | 6,683 | 0 | 0 | 0 | 0 |
| G | 1,792 | 0 | 0 | 0 | 0 |

Next are the statistics by the individual case and their unweighted average.

Baseline Breakdown by Case

| Case | Precision | Recall | F-score |
|---|---|---|---|
| N | 38.26% | 100.00% | 55.35% |
| A | 0.00% | 0.00% | 0.00% |
| D | 0.00% | 0.00% | 0.00% |
| G | 0.00% | 0.00% | 0.00% |
| Average | 9.57% | 25.00% | 13.84% |

The following table summarizes the performance, taking into account nouns left unmarked by the algorithm. Here, the numbers are the same for both columns since all nouns are marked.

Baseline Overall Results

| Result | Count | Of marked nouns | Of all nouns |
|---|---|---|---|
| Total marked correctly | 9,200 | 38.26% | 38.26% |
| Total marked incorrectly | 14,844 | 61.74% | 61.74% |
| Total left unmarked | 0 | N/A | 0.00% |
| Total wrong | 14,844 | N/A | 61.74% |

It is unsurprising that the baseline is correct exactly as often as nominative occurs in the corpus. The other two baseline algorithms did not fare as well and therefore are not reported in full. The algorithm that marks cases randomly in proportion to their frequencies assigned the correct cases approximately 28% of the time, and the baseline of assigning case randomly (each case getting a 25% chance of being assigned) was correct about 25% of the time, as one would expect.

## 8.3 Lexical Step

In order to measure "to what degree" Icelandic is quirky, I present the results of running just the first part of the lexical step (1a) and the portion of the second part (1b) that assigns case to objects of prepositions (excluding the part of 1b that assigns dative to indirect objects and genitive to possessors). This substep is run as part of both the GFBA and SBA because they make the same predictions about quirky verbs' arguments.

Since it given the advantage of never marking the wrong case (described in Section 6.6), there is no need for a confusion matrix or a breakdown by case beyond reporting the raw numbers. Nominative is never lexically specified, as discussed in Section 6.5. There were 3,032 nouns assigned accusative in this substep, 4,544 assigned dative, and 497 assigned genitive. Those numbers leave 15,971 nouns lexically unspecified.

Lexical Step Overall Results

| Result | Count | Of marked nouns | Of all nouns |
|---|---|---|---|
| Total marked correctly | 8,073 | 100.00% | 33.58% |
| Total marked incorrectly | 0 | 0.00% | 0.00% |
| Total left unmarked | 21,411 | N/A | 66.42% |
| Total wrong | 21,411 | N/A | 66.42% |

These results show that lexically-governed case is a significant, though not overwhelming, portion of the algorithm. That being said, the prepositions (approximately 32.5%) contribute much more than the verbs (approximately 1%).

## 8.4 Grammatical-Function–Based Algorithm

This section includes the results of running Step 1 of the SBA (both part a's quirky items and part b's prepositions, dative "applicatives"/indirect objects, and genitive possessors) and then the GFBA, all as described in Section 7.3. Part b is included because it has access to the case frames of prepositions that the GFBA's steps do not, while its assignment of dative and genitive are identical to the GFBA' s and are therefore benign (order of assignment of those doesn't matter). Once those approximately 2,600 nouns are marked as part of the lexical step, the function-based assignments proceed, generating the following results.

GFBA Confusion Matrix

| Case assigned by algorithm → / Case listed in corpus ↓ | N | A | D | G | Unmarked |
|---|---|---|---|---|---|
| N | 7455 | 66 | 52 | 134 | 1493 |
| A | 262 | 5396 | 83 | 254 | 374 |
| D | 418 | 625 | 5066 | 199 | 375 |
| G | 7 | 74 | 19 | 1598 | 94 |

GFBA Breakdown by Case

| Case | Precision | Recall | F-score |
|---|---|---|---|
| N | 91.56% | 81.03% | 85.98% |
| A | 87.58% | 84.72% | 86.13% |
| D | 97.05% | 75.8% | 85.12% |
| G | 73.14% | 89.17% | 80.36% |
| Average | 87.33% | 82.68% | 84.40% |

These results include unmarked nouns in the calculation of recall. If however, these are excluded, the algorithm's average f-score becomes about 88%.

GFBA Overall Results

| Result | Count | Of marked nouns | Of all nouns |
|---|---|---|---|
| Total marked correctly | 19515 | 89.90% | 81.16% |
| Total marked incorrectly | 2193 | 10.1% | 9.12% |
| Total left unmarked | 2336 | N/A | 9.72% |
| Total wrong | 4529 | N/A | 18.84% |

It is worth remarking that the lexical step gives a substantial boost to the GFBA's performance. Without any of Step 1, the algorithm only marks 66.22% of all nouns correctly.

## 8.5 Structure-Based Algorithm

To give a sense of how much work is being left to the default step, it is excluded from these results, though the SBA's scores are reported with the default after each table.

SBA Confusion Matrix

| Case assigned by algorithm → / Case listed in corpus ↓ | N | A | D | G | Unmarked |
|---|---|---|---|---|---|
| N | 7,260 | 1290 | 54 | 145 | 451 |
| A | 531 | 5,406 | 116 | 263 | 53 |
| D | 551 | 832 | 5,081 | 209 | 10 |
| G | 37 | 129 | 20 | 1,601 | 5 |

To calculate the results with the default step, one simply adds the last column of the confusion matrix to the first. The resulting first column is 7,711; 584; 561; 42. This change leaves zero nouns unmarked and does not affect the middle three columns.

SBA Breakdown by Case

| Case | Precision | Recall | F-score |
|---|---|---|---|
| N | 86.65% | 78.91% | 82.6% |
| A | 70.6% | 84.88% | 77.09% |
| D | 96.4% | 76.03% | 85.01% |
| G | 72.18% | 89.34% | 79.85% |
| Average | 81.46% | 82.29% | 81.14% |

When the default step's assignments are included, the average f-score is 81.79%.

SBA Overall Results

| Result | Count | Of Marked Nouns | Of all nouns |
|---|---|---|---|
| Total marked correctly | 19348 | 82.24% | 80.47% |
| Total marked incorrectly | 4177 | 17.76% | 17.37% |
| Total left un-marked | 519 | N/A | 2.16% |
| Total wrong | 4696 | N?A | 19.53% |

When the default is included, then the total percentage correct of all nouns becomes 82.34%.

It is informative to examine the results without the default, but the results discussed in the following section will all include the default since it is properly part of the SBA.

# 9 Discussion

## 9.1 Summary of results

| Algorithm | Raw score on all nouns | Raw score on only marked nouns | Average f-score on all nouns |
|---|---|---|---|
| Baseline (all NOM) | 38.26% | 38.26% | 13.84% |
| GFBA (with quirky items) | 81.16% | 89.90% | 84.40% |
| SBA (with quirky items and default) | 82.34% | 82.34% | 81.79% |

## 9.2 Comparing the results

Overall, the two algorithms fare quite similarly, but for different reasons. The mechanisms for assigning quirky case, dative case, and genitive are identical or nearly identical across the two implementations, contributing to their similarity. However, their assignment of accusative and nominative is different but nonetheless results in a similar percentage of correctly assigned nouns.

The GFBA makes substantially fewer incorrect assignments of both nominative and accusative, and it rarely marks a noun that is nominative in the corpus as anything else. However, there are many nouns that are nominative and accusative (and dative, too) in the IcePaHC that the GFBA leaves unmarked. The fact of the matter is that many nouns are just not subjects or objects.

In contrast, the SBA leaves no nouns unmarked, but misfires much more often. In particular, it often over-assigns accusative to nouns that are nominative in the IcePaHC and dative to nouns that should be accusative. The structures that create these incorrect assignments are discussed below in Section 9.4.

One important consideration is that the grammatical functions as labeled in the tree bank are close to perfect for the GFBA's purposes. The GFBA's scores therefore benefit from this strong compatibility. On the other hand, the IcePaHC trees are very flat, and the internal structure of IPs is not nearly as detailed as the one assumed by McFadden (2004). These differences make it difficult for the SBA to have access to all the information required for it to work as described in theory.

For this reason, one must take the SBA's results (and thereby their similarity to the GFBA's results) with a grain of salt. In light of the limitations discussed in Sections 6 and 7, it is not implausible that the SBA would fare better on a tree bank whose structure is more suited to the distinctions it makes.

## 9.3 Known exceptions in theory

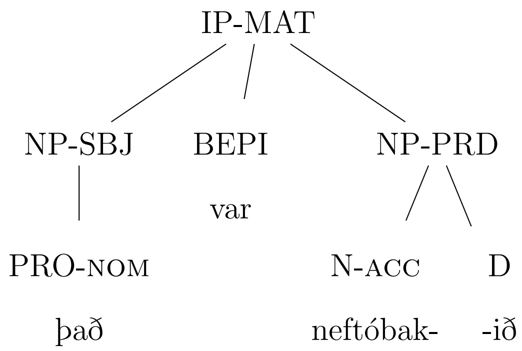
There are a few known exceptions to the two algorithms (in theory) that are worth mentioning. A common one for SBA is noun predicates. In reality, the predicate should agree in case with the subject, but Step 2 generally assigns accusative case to the lower noun, regardless of what the higher one's case ends up being. There are also some NP-internal dative possessors, in contrast with the expected genitive inside NPs.

While (apparently) infrequent, the GFBA makes some known systematic errors as well. In ECM constructions, for instance, the subject of the embedded clause bears accusative, not nominative. In addition, when a quirky verb lexically specifies its subject's case, the object generally bears nominative, not accusative.

## 9.4 Patterns of error in practice

For the SBA, assigning accusative instead of nominative and dative are the two most frequent errors by a fair margin. One very common structure that causes this error is the known problem of predicates. When the subject is unmarked, the predicate will be locally c-commanded (and preceded, as required by Section 6.11) by the subject and therefore receive dependent accusative instead of agreeing in case with the subject, which later frequently receives unmarked nominative. An example of this phenomenon is given below. (This sentence glosses to "That was the snuff.")

(9) það var neftóbakið




In several other instances, adverbial NPs (along the lines of "bit by bit") that immediately precede subjects (and sometimes objects) also license dependent accusative where it shouldn't be assigned (both on nouns that should be nominative and dative). Though the IcePaHC parses these licensing nodes as NPs, they act as adverbs; if McFadden treats them as such, it would explain why they don't appear to license accusative on their siblings in the tree bank. This effect is most frequently observed where the case should be nominative, but it also occurs for should-be dative nouns. That fact is likely because nominative is simply the most frequent case, especially for subjects, which commonly occur as siblings of the adverbial NPs.

Aside from the unmarked nouns, the only type of error that the GFBA makes more than 500 times is assigning accusative instead of dative. These assignments occur primarily when a direct object bearing dative occurs inside embedded (often infinitival) IPs. That said, the leaving nouns that should be nominative unmarked is by far the most frequent "error" that the GFBA makes. Once again, I conclude that many NPs simply don't occur as subjects, objects, or possessors.

It is easy to estimate the error introduced by the fact that, as described in Section 6.9, pronominal quantifiers are marked with case but do not always projection QPs (or NPs) that are marked with grammatical functions and, as a result, will not always be assigned case by the GFBA. These account for approximately 50 of the "unmarked" errors made by the GFBA (though they also make up about 100 correctly marked nouns) and are therefore not a significant concern.

## 9.5 Number and Diversity of Sources

Though there are only four sources from the past 30 years, all of which are novels, they collectively contain nearly one hundred thousand words. While having few sources is an unfortunate limitation of what resources are available, the diversity of the sources can be coarsely checked by scoring the models separately on each source. Indeed, the performances on each of the four sources individually are comparable, from which we conclude that the sources are more likely (though far from certain) to be representative of Icelandic as a whole.

## 9.6 Structure and Function Revisited

The historical appeal of the GFBA together with its strong performance in the instances where a case was assigned make an argument in its defense. That is, there is certainly *some* relationship between case and grammatical function. To an extent, the GFBA's good performance may result from the fact that the grammatical functions within the corpus are labeled with much more detail than the structure is, but nonetheless, its results stand. Perhaps the SBA would have done better with a more detailed corpus, but the GFBA still scored 90% correct when it knew what to do with a noun (i.e. excluding unmarked nouns from the denominator).

As such, I wonder whether the future of Case Theory might lie in an integrated approach that draws from both structure and function, which are intimately related as a central tenant of syntax. While this approach is not justified by any theory, the following results from a "Franken-algorithm" woven together from the successful pieces of the SBA and GFBA suggest that integrating the two is an idea with exploring.

I tried my hand at one by merging the two algorithms into a new one that first runs the lexical step, both parts a (quirky items) and b (dative applicatives and objects of prepositions, and genitive possessors ), then runs the GFBA, and finally runs the rest of the SBA on the nouns that the modified GFBA leaves unmarked. This algorithm will be limited by many of the same challenges that the SBA faces (see Section 6). While the principles behind this combination of steps are likely not theoretically coherent, the resulting score is slightly higher than both individual algorithms'. It marks case correctly 86.78% of the time and has an average f-score of 85.09%.

# 10 Next Steps

Case has not been solved yet. There is more analysis to be had of these algorithms, particularly looking at where the two disagree and which (if either) is right in those instances. There are also several questions that go unanswered, and this project creates flexibility to probe the resources at hand in ways that can answer questions beyond the scope of this paper. Some of these questions or areas to be explored are listed below. I invite the use of my code (available in Appendix A) to answer these and others.

## 10.1 Quirky lexical items

1. How often are the different grammatical functions (subjects/objects/etc.) lexically specified, both in absolute and compared to each other?

2. Look at precision, recall, and f-score by function (subjects, direct objects, etc.).

3. How "good" are the lists of quirky verbs? Can one reverse-engineer the (most frequent) frames of known quirky verbs? Of all verbs?

4. When lexically specified items don't find arguments, are the arguments there or not? There are several ways an argument could be "somehow null:" non-noun complements, missing/no arguments, and empty or elided arguments. On the other hand, it is also possible that the program simply does not find the arguments that are there.

## 10.2 Empirical case-marking

1. What strengths and weaknesses of the algorithms are there on the level of granularity of individual steps? (That is, identify individual steps that work particularly well or poorly.)

2. What do these results say about the distribution of case in Icelandic? Are there generalizations to be made?

3. Has the distribution of case in Icelandic changed diachronically? (Are the results of one or more algorithms significantly different on earlier documents)

4. What more general correlations are there between structures, functions, and case?

## References


[1] Barðdal, Jóhanna. 2011. The rise of Dative Substitution in the history of Icelandic: A diachronic construction grammar account. *Lingua* 121:60-79.

[2] Butt, Miriam. *Theories of Case*. Cambridge, UK: Cambridge UP, 2006. Print.

[3] Jónsson, Jóhannes Gísli. 2000. Case and double objects in Icelandic. *Leeds Working Papers in Linguistics and Phonetics* 8:71-94.

[4] Jónsson, Jóhannes Gísli. Not so quirky: On subject case in Icelandic. *New Perspectives on Case Theory*, 127-163. Stanford, CA: CSLI, 2003.

[5] Jónsson, Jóhannes Gísli. *Verb-classes and dative objects in Insular Scandinavian.* Philadelphia, PA: Benjamins, 2009. Print.

[6] Maling, Joan and Jóhannes Gísli Jónsson. 1995. On Nominative Objects in Icelandic and the Feature [+Human]. *Working Papers in Scandinavian Syntax* 56:71-5.

[7] Marantz, Alec. 2000. Case and Licensing. *Arguments and Case* 11-30.

[8] McFadden, Thomas. 2004. The Position of Morphological Case in the Derivation: A Study on the Syntax-Morphology Interface. Diss. U of Pennsylvania. Version 0.9. http://www.linguist.is/icelandic$_t$reebank

[9] Sigurðsson, Halldór. 2012. Minimalist C/case. *Linguistic Inquiry* 43:191–227.

[10] Tebbutt, John. 1995. "Mimir – Icelandic Grammar Notebook." N.p., 1995. Web. 1 Feb. 2015. <http://www.samkoma.com/mimir/mimir.htm>.

[11] Wallenberg, Joel C, Anton Karl Ingason, Einar Freyr Sigurðsson and Eiríkur Rögnvaldsson. 2011. Icelandic Parsed Historical Corpus (IcePaHC).

[12] Wood, Jim. 2011. Icelandic *let*-causatives and case. *Working Papers in Scandinavian Syntax* 87:1-52.

[13] Wood, Jim and Halldór Ármann Sigurðsson. 2014. *Let*-causatives and (a)symmetric DAT-NOM constructions. *Syntax* 17, 3:269-298.

[14] Wood, Jim. *Icelandic Morphosyntax and Argument Structure*. Dordrecht: Springer, 2015. Print.

[15] Yip, Moira, Joan Maling, and Ray Jackendoff. 1987. Case in Tiers. *Language* June 1987, 63(2):217-250.

[16] Zaenen, Annie, Joan Maling, and Höskuldur Thráinsson. 1985. Case and grammatical functions: The Icelandic passive. *Natural Language and Linguistic Theory* 3:441-483.

# A The Program

The most up-to-date code of the program is publicly available at `https://github.com/milescalabresi/Ling-Thesis`.

# B Lexical Information

The following is the list of "quirky" verbs and prepositions, together with the given case frames from Jónsson (2000), Jónsson (2003), Jónsson (2009), Barðdal (2011), and Tebbutt (1995). This list is slightly condensed in that multi-word expressions are collapsed to only the first word (see Section 6.10). Any case frames associated to an expression that begins with that word are associated with that word in the lexical step.

## B.1 Verbs

ægja: D--

æsa: -AG

æskja: --G

æsta: -AG

afklæða: -AD

afla: --G

aflétta: D--

afsala: --D

aga: D--

akka: D--

árna: --G

áskotnast: D--

auðnast: D--

bæta: --G

bætast: D--

baga: A--

batna: D--

beiða: -AG

beina: --D

beita: -AD

berast: D--

biðja: -AG

bjáta: D--

bjóða: D--

bjóðast: D--

blæða: D--

bláskarða: A--

blöskra: D--

blóta: --D

bóla: A--

bólstra: A--

bora: A--

bragðast: D--

bresta: A--

brima: A--

brjóta: A--

brúa: A--

búnast: D--

byrja: D--

daga: A--

dáma: D--

daprast: D--

demba: D--

deprast: D--

detta: D--

dirfa: A--

doðra: D--

dotta: A--

drátta: A--

drepa: A--

dreyma: A--

drífa: A--

duga: D--
dveljast: D--
dylja: -AG
dyljast: D--
eggja: -AG
eima: A--
elda: D--
endast: D--
eymast: D--
fá: --G
fæðast: D--
falla: D--
fara: D--
farast: D--
farða: A--
farnast: D--
fata: D--
fatast: D--
fatlast: D--
fatra: D--
feila: D--
fénast: D--
fenna: A--
fiðra: A--
finna: A--
finnast: D--
fipast: D--
firna: A--
firra: -AD
fjölga: D--
flæða: A--
fletja: A--
fletta: -AD
fleyta: D--
fljúga: D--
flökra: A--
förla: D--
förlast: D--
fórna: --D
forvitna: A--
frábiðja: --D
fregna: -AG
frétta: -AG
frýja: --G
furða: A--
fylgja: D--
fylla: A--
fyrirgefast: D--
fyrirkunna: -AG
fyrirmuna: --G
fýsa: A--
gæta: G--
gagnast: D--
ganga: D--
geðjast: D--
gefa: D--
gera: D--
gerast: G--
geta: --G
getast: D--
gifta: -AD
girna: A--
gleðja: A--
gola: A--
græðast: D--
greina: A--
gremjast: D--
grípa: A--
gruna: A--
hæfa: D--
hægja: D--
hætta: D--
haga: D--
halda: D--
haldast: D--
halla: D--
harka: D--
hasa: A--
hefja: A--
hefna: --G
hefnast: D--
heilsast: D--
heita: --D
héla: A--
helga: -AD
hema: A--
henda: A--
henta: D--
heppnast: D--
heyrast: D--
hilla: A--
hissa: A--
hita: A--
hitna: D--
hjóma: A--
hlaða: D--
hlekkjast: D--
hleypa: D--
hlotnast: D--
hlýða: D--
hlýna: D--
hnigna: D--
hnita: A--
hnota: A--
hnykkja: D--
hnýta: A--
höfga: D--
hóta: --D
hraka: D--
hreistra: A--
hrekja: A--
hríma: A--

hrökkva: D--
hrolla: D--
hryggja: A--
hrylla: A--
hugkvæmast: D--
hugnast: D--
hugsast: D--
hungra: A--
hverfa: --G
hvetja: -AG
hviðra: A--
hvolfa: D--
iðra: A--
játa: --D
kæfa: A--
kala: A--
kálfa: A--
káma: A--
kefja: A--
kemba: A--
kenna: --G
kilpa: A--
kitla: A--
klæða: A--
klæja: A--
klígja: A--
kólna: D--
koma: D--

kopa: D--
krefja: -AG
kreppa: A--
kunna: -AG
kylja: A--
kyngja: D--
kynja: A--
kyrra: A--
lægja: A--
lærast: D--
lánast: D--
langa: D--
látt: D--
láta: D--
latra: D--
legast: D--
leggja: A--
leggjast: D--
leiðast: D--
leita: --G
lemja: A--
lenda: D--
lendast: D--
lengja: A--
lengjast: D--
letja: -AG
létta: D--
léttast: D--

leyfast: D--
leyna: -AD
leysa: A--
líða: D--
líðast: D--
liggja: D--
liggjast: D--
líka: D--
lina: D--
linna: D--
lítast: D--
ljá: --G
ljósta: D--
ljúka: D--
lofa: --D
lykta: D--
lynda: D--
lysta: A--
mælast: D--
marka: -AD
mega: A--
miða: D--
miðla: --D
minna: A--
misfarast: D--
misheppnast: D--
misheyrast: D--
mislíka: D--

misminna: A--
missýnast: D--
mistakast: D--
misunna: --G
mjaka: D--
næða: A--
nægja: D--
nálægja: -AD
nátta: A--
nauðsynja: A--
nepja: A--
notkast: D--
núa: --D
nýtast: D--
óa: D--
ofbjóða: D--
offerja: D--
offra: --D
ofhasa: A--
óhaga: A--
ómætta: A--
ona: A--
opnast: D--
óra: A--
orka: --G
orna: D--
örvænta: --G
óska: --G

ota: D--
ótta: A--
passa: D--
ræna: -AD
raga: A--
ráma: A--
rámka: A--
rangminna: A--
redda: --D
reiknast: D--
reynast: D--
ríða: D--
rífa: A--
rigna: D--
rjúfa: A--
rofa: A--
ryðja: --G
rýja: -AD
sækja: A--
sækjast: D--
sæma: D--
sæta: D--
sáldra: D--
samhæfa: -AD
samlaða: -AD
samlaga: -AD
samlíkja: -AD
samræma: -AD
samræta: -AD
samrýma: -AD
samsama: -AD
samtengja: -AD
samtvinna: -AD
sárna: D--
segja: --D
seinka: D--
semja: D--
setja: A--
sigra: A--
sinnast: D--
sjá: A--
sjást: D--
skæna: A--
skafa: A--
skálda: A--
skána: D--
skarða: A--
skeika: D--
skeipla: D--
skella: D--
skemma: A--
skera: A--
skilja: A--
skiljast: D--
skirra: -AD
skjátlast: D--
skjótast: D--
skola: D--
skorta: A--
skuldskeyta: -AD
slá: D--
slaka: --D
slengja: D--
sleppa: D--
slota: D--
slysast: D--
smjúga: A--
snara: A--
snerpa: A--
sneyða: -AD
snjóa: D--
snúðgast: D--
snugga: A--
sofnast: D--
sóma: D--
spá: --D
spyrja: -AG
stafa: A--
standa: D--
stansa: A--
stela: -AD
stemma: A--
steypa: D--
stíra: A--
stugga: A--
sundla: A--
svala: D--
svarfla: D--
svelgjast: D--
svengja: A--
svía: D--
svíða: D--
svíkja: -AD
svima: A--
svipast: D--
svipta: -AD
syfja: A--
sýla: A--
sýnast: D--
synja: --G
taka: A--
takast: D--
talast: D--
teljast: D--
tengja: -AD
þarfna: D--
þoka: D--
þóknast: D--
þrjóta: A--
þrota: A--
þurrka: A--
þvera: A--

þverra: A--
þykja: D--
þyngja: D--
þyngjast: D--
þyrsta: A--
tíða: A--
tilfalla: D--
tíma: A--
tólga: A--
tyrma: D--
ugga: A--
undanskilja: -AD
undanþiggja: -AD
undirgefa: -AD
undra: A--
unna: --G
usla: A--
útdeila: --D
úthluta: --D
vægja: D--
væma: A--

valda: --D
vanhaga: A--
vanka: A--
vanta: A--
vara: A--
varða: A--
varna: --G
vegna: D--
veita: D--
veitast: D--
velgja: A--
velkja: A--
venda: D--
venja: -AD
verja: -AD
verkja: A--
versna: D--
víkja: --D
vinda: D--
vinnast: D--
vippa: D--

virðast: D--
vitnast: D--
volgna: D--
vona: A--
yfirþyrma: A--
ylna: D--
ysta: A--
bera: DAD, A--
birta: A-- , D--
bíta: A-- , D--
blása: A-- , --D
bregða: -DD , D--
draga: A-- , D--
fækka: -D- , D--
festa: A-- , D--
fleygja: D-- , --D
kveðja: --G , -AG
langa: A--, D--
muna: A-- , D--
nema: --G , -AD
óa við: A--, D--

offerja: A--, D--
ógna: D-- , D-D
reiða: A-- , D--
reka: A-- , A-G
renna: A-- , D--
saka: A-- , AAG
skila: D-- , --D
skipta: -AD , --D
skjóta: D-- , --D
slíta: A-- , D--
svíða: A--, D--
svipa: A-- , D--
væna: --D, -AG
væna: --G , -AG
vænta: A-- , --G
vera: D-- , G--
verða: D-- , G--

## B.2 Prepositions

á: --A, --D

að: --D

af: --D

án: --G

andspænis: --D

ásamt: --D

auk: --G

austan: --G

eftir: --A, --D

frá: --D

fyrir: --A, --D

gagn: --D

gagnvart: --D

gegnt: --D

gegnum: --A

handa: --D

hjá: --D

í: --G, --A, --D

innan: --G

kringum: --A

með: --A, --D

meðal: --G

meðfram: --D

megin: --G

milli: --G

millum: --G

mót: --D

móti: --D

nálægt: --D

neðan: --G

norðan: --G

ofan: --G

sakir: --G

sökum: --G

sunnan: --G

til: --G

um: --A

umfram: --A

umhverfis: --A

undan: --D

undir: --A, --D

úr: --D

utan: --G

vegna: --G

vestan: --G

við: --A, --D

yfir: --A, –D