Abstract

We describe an empirical method to explore and contrast the roles of default and principal part information in the differentiation of inflectional classes. We use an unsupervised machine learning method to classify Russian nouns into inflectional classes, first with full paradigm information, and then with particular types of information removed. When we remove default information, shared across classes, we expect there to be little effect on the classification. In contrast when we remove principal part information we expect there to be a more detrimental effect on classification performance. Our data set consists of paradigm listings of the 80 most frequent Russian nouns, generated from a formal theory which allows us to distinguish default and principal part information. Our results show that removal of forms classified as principal parts has a more detrimental effect on the classification than removal of default information. However, we also find that there are differences within the defaults and principal parts, and we suggest that these may in part be attributable to stress patterns.

1. Introduction

The particular challenge which languages with inflectional classes pose is that these classes create an additional layer of complexity which is more or less irrelevant from the perspective of syntax. Linguists can provide principled analyses of such inflectional classes, and typically have a good idea of what the main ones in a language are. However, our understanding of inflectional classes could be improved by exploring how well linguistically informed analyses correspond to those which are obtained using unsupervised learning techniques, with few built-in assumptions. This would provide some external validation for such analyses.

We need first to be clear about the way in which inflectional classes are complex. They represent a particular kind of morphological complexity which it is important to distinguish from other phenomena which may be associated with these terms. Consider the Turkish verb in (1), discussed by Baerman et al. (2009).

(1) alıyorduysam
al-ıyor-du-isa-m
take-CONTINUOUS-PST-CONDITIONAL-1SG
‘if I was taking’

Here a large number of inflectional suffixes are attached to the root. But this large number is a direct reflection of the distinctions relevant for syntax. So this is no more complex than the underlying requirements of syntax and is therefore quite straightforward. In Figure 1, in contrast, there is complexity in Russian nouns arising solely from membership of inflectional classes with no corresponding syntactic requirement.[1]

	‘deed’	‘factory’	‘country’	‘bone’
	Class IV	Class I	Class II	Class III
NOM SG	del-o	zavod	stran-a	kost′
ACC SG	del-o	zavod	stran-u	kost′
GEN SG	del-a	zavod-a	stran-i	kost′-i
DAT SG	del-u	zavod-u	stran-e	kost′-i
PREP SG	del-e	zavod-e	stran-e	kost′-i
INS SG	del-om	zavod-om	stran-oj	kost′-ju
NOM PL	del-a	zavod-i	stran-i	kost′-i
ACC PL	del-a	zavod-i	stran-i	kost′-i
GEN PL	del	zavod-ov	stran	kost′-ej
DAT PL	del-am	zavod-am	stran-am	kost′-am
PREP PL	del-ax	zavod-ax	stran-ax	kost′-ax
INS PL	del-am′i	zavod-am′i	stran-am′i	kost′-am′i

Figure 1: Russian inflectional classes (phonological transcription)[2]

This complexity cannot be explained by the role of gender assignment. The words *strana* ‘country’ and *kost’* ‘bone’, for example, both require feminine gender on agreeing items, but inflect differently. On the other hand, the words *delo* ‘deed’ and *zavod* ‘factory’ require different gender agreement (neuter and masculine respectively), but share many inflections in the singular, while all the classes share many inflections in the plural. Hence, the relationship between the noun inflectional classes (IV, I, II and III) and gender is not a direct one. Gender is relevant for syntax, as it is an agreement category. Inflectional class, on the other hand, is not relevant for syntax, as

[1] We have placed IV to the left of I in Figure 1, because they can be treated as belonging to a superclass (see Corbett and Fraser, 1993).

[2] The phonolological transcription assumes that /i/ has two allophonic variants. It is retracted to the allophone [ɨ] after non-palatalized consonants. The nominative plural form /zakoni/, for example, will be realized with [ɨ], but /kost'i/ retains [i] since [t'] is soft. An automatic rule of palatalization applies before the vowel /e/. The marker ′ indicates that a consonant is palatalized.

the distinction between class II and III for example has no ramifications in the rules of agreement. This is pure morphological complexity whereby one and the same grammatical distinction can be expressed in a number of different ways. This is additional structure which is not relevant from the point of view syntax. In other words, it is complexity associated with autonomous morphology in the sense of Aronoff (1994).

1.1 Defaults and principal parts

The question naturally arises therefore as to what makes morphological complexity of this type learnable. Two theoretical notions can be mustered when describing the properties of inflectional classes. One is the traditional notion of *principal part*. This is the form, or set of forms, which make it possible to infer the other forms of a lexeme. The other notion is *default*. Finkel and Stump (2010) define the canonical principal part as both highly predictive and highly unpredictable. That is, given a canonical principal part we can predict all the other forms in a lexeme's paradigm. Conversely, the other forms in the paradigm would not predict a canonical principal part. Using this terminology we can see that a default is the mirror image of this. A canonical morphological default is a form which does not serve to predict the other forms in a lexeme's paradigm, but is highly predictable (in the limiting case because all lexemes have it).

As is clear from Figure 1 some items should be good as principal parts for identifying their inflectional class, whereas others are defaults. There are good theoretical grounds for assuming that, at some level, Russian has four nouns inflectional classes. If we analyze Russian declensions as a default inheritance hierarchy, we can treat certain classes, such as I and IV, as belonging to a superclass (labelled N_O by Corbett and Fraser 1993 in their Network Morphology analysis).

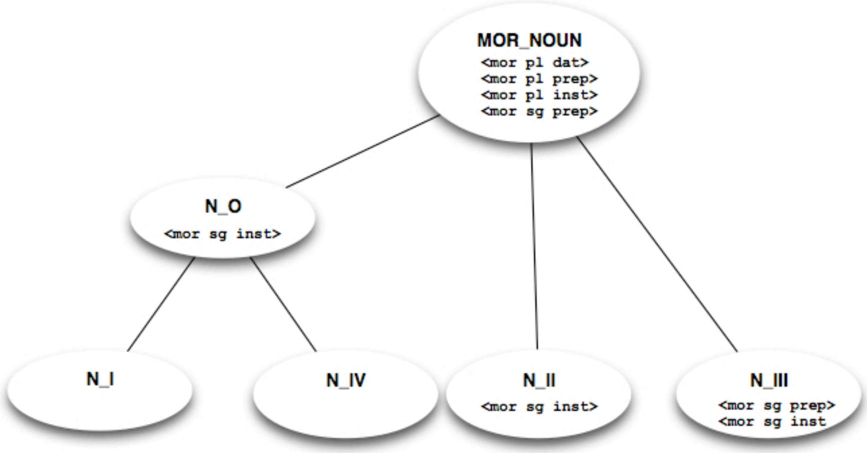

Figure 2: defining defaults and principal parts in terms of inheritance

In Figure 2 we consider 6 of the 12 paradigm main paradigm cells for Russian in terms of the notion principal part and default. We give the locations where something has to be said about the realization of these 6 cells. (We do not give any information about the 6 other paradigm cells in Figure 2.) The paradigm cells plural dative, plural instrumental and plural prepositional (represented by the paths `<mor pl dat>`, `<mor pl inst>`, `<mor pl prep>`) are the most default-like, because they are not overridden by any of the lower nodes.[3] The rules which define them are therefore located at the highest node only. Knowing the plural dative, prepositional or instrumental is of no help in inferring the other forms in the paradigm of a given noun. On the other hand, they are predictable. We can have the highest degree of certainty about what a noun's plural dative, prepositional and instrumental will look like. Examination of Figure 1 shows that we can be fairly certain about the singular prepositional inflection of a noun. It is only class III which has a different realization for this, and this is reflected in the fact that something (`<mor sg prep>`) needs to be stated at `N_III` about the singular prepositional.

The singular instrumental (`<mor sg inst>`), on the other hand, has to be stated at three locations (`N_O`, `N_II` and `N_III`). Knowing the singular instrumental is more helpful in facilitating prediction of other forms, although it will not distinguish between class I and class IV. The singular prepositional is therefore more default-like than the singular instrumental, which we can consider more principal-part-like. Given the analytical decisions taken to place defaults at different points in the hierarchy (e.g. Corbett and Fraser 1993; Brown et al. 1996; Baerman, Brown and Corbett 2005; Brown and Hippisley forthcoming) we can test to see whether there is a reflex in the learning of inflectional classes by systematic removal of information. We can compare the default-like with the principal-part-like information (the latter being located lower in the hierarchy at the declension class nodes, as with the singular instrumental).

1.2 Classification, defaults and principal parts

In this paper we explore how well an unsupervised learning method classifies nouns into inflectional classes, and consider the degree to which these classes match with ones which have been identified for Russian. The ability to classify the items must rely on information from the paradigm cells, but only with systematic testing can we determine which information plays a significant role. Given that the classification must be based on paradigm cell

[3] Figure 2 is actually a simplification in that the plural dative, instrumental and prepositional are defaults at the `MOR_NOMINAL` level, because the rules associated with them can generalize over the other nominal classes (such as adjectives and pronouns). This is discussed in Brown and Hippisley (forthcoming).

information, it is a task which is related to what Ackerman *et al.* (2009) call the Paradigm Cell Filling Problem (PCFP):

> "What licenses reliable inferences about the inflected (and derived) surface forms of a lexical item?"
> (Ackerman *et al.* 2009: 54)

Ackerman *et al.* (2009) claim that the tractability of this problem is guaranteed by the fact that inflectional classes are constrained to reduce entropy, so that not all instances of particular inflectional exponents are equally probable. Finkel and Stump (2007) appeal to the traditional notion of principal parts so as to reduce the entropy down to zero. Paradigm cells such as the instrumental singular appear to be very informative as to class. The underlying analysis with which we have created the dataset for the experiments has itself a gradient notion of default. We have other defaults which have an intermediate status, as with the singular prepositional. For example, knowing the nominative plural narrows down the set of possible classes (I-III). And a default may sometimes even help distinguish between classes. This is true for the nominative plural in that class I has the default form, while class IV does not. Our aim, then, is to determine what role these different notions play in the unsupervised learning of inflectional classes. The work we present here is an initial step towards understanding this.

The ideal unsupervised method should be quite robust and independent of format, with very few theoretical assumptions built in. Goldsmith and O'Brien (2006) use a feed-forward backpropagation neural network with one hidden layer to simulate the learning of Spanish conjugation classes. The hidden layer allows for a better classification into these classes. They also simulated the acquisition of German noun declensions using this method. The method we use is relatively independent of data format and does not make use of a hidden layer. There are, of course, some issues with it, which we discuss in section 2.1.

We apply our chosen unsupervised learning method to full paradigms generated from an underlying default-based theory of Russian. This allows us to test how well linguists' intuitions about inflectional classes fare when tested with few built-in assumptions. We use the full paradigms of the 80 most frequent noun lexemes from Zasorina's (1977) frequency dictionary. This allows us to consider how readily inflectional class membership can be inferred from high frequency data, where that are lots of items which appear to be fuzzy or partial members of a class. We can then see how well the classification performs by removing default and principal parts information.

An additional complication to our task is that stress patterns play a role in Russian noun inflection, and these cross-classify the noun declension. The task of inferring an inflectional class and the appropriate stress pattern results is a greater challenge. Combined with the fact that there is a rich tradition of research on Russian to draw from, this additional complexity makes the language an important testing ground for methods for inferring and validating inflectional classes. Particularly among high frequency nouns, there are items which may have the right affixes for a particular inflectional class but stress patterns which may associate them with nouns which belong to another inflectional class, or certain cells of a nouns' paradigms have affixes which are not typical for the class with which they are best associated. We are currently working on separating out the role of the stress patterns from the declensions, and will not discuss this in great detail in this paper.

2. Unsupervised learning of inflectional classes

Our empirical investigation of these notions from morphological theory employs an unsupervised machine learning technique to derive inflectional classes from sets of noun paradigm tables. We use compression-based similarity to cluster nouns into classes, where nouns in the same class are considered to have more similar paradigm tables than nouns in different classes. The core of our method is CompLearn[4], a machine-learning system which relates arbitrary data objects according to their 'similarity' (section 2.1). However, CompLearn does not implement the actual clustering of similar data into classes, so we need to introduce some simple heuristics to achieve this additional step (section 2.2). These two components provide the basic framework for a method for learning inflectional classes. We discuss methods for evaluating the results of the learning task (section 2.3), and finally summarise the complete experimental method (section 2.4).

2.1 Compression-based machine learning

The machine-learning paradigm that we use is the compression-based approach described in Cilibrasi and Vitányi (2005) and Cilibrasi (2007), as implemented in the CompLearn tools. This approach has two main components: (a) the use of compression (in the sense of standard compression tools such as zip, bzip etc.) as the basis of a measure for comparing data objects and (b) a heuristic clustering method, which relates objects according to their similarity using this measure. Together, these components provide a general purpose unsupervised method for clustering arbitrary digital data objects. Cilibrasi (2007) provides examples of its application to fields as diverse as genetics in mammals and viruses, music, literature, and genealogical relatedness of languages.[5]

[4] http://www.complearn.org

[5] Other work using compression-based techniques in relation to the study of language includes Juola (1989) and Kettunen et al. (2006). This research focused on compressing corpus data. While Juola's work addresses morphology, it is concerned with measuring complexity in terms

The basic operation of the CompLearn method is as follows. The input to the system is a set of data objects, each of which is simply a computer file containing some (unconstrained) digital data. Given two such data objects, CompLearn determines how similar they are by calculating the *normalized compression distance* (NCD) between them. This exploits the notion of a *compression function* which attempts to make a data object smaller by detecting repeated patterns in the data and representing them more compactly (as commonly used by computer operating systems to reduce the size of large files). NCD measures the difference between data objects by comparing how well they compress jointly and separately – if there is a benefit to compressing them jointly, this must be because the compression algorithm has found commonalities between them, and we interpret this as meaning they are similar. The more benefit that is gained, the more similar the two data objects are.

Given two data objects *x* and *y* and a compression function *c*, NCD is defined as:

$$(2) \qquad NCD(x,y) = \frac{C(xy) - \min\{C(x),C(y)\}}{\max\{C(x),C(y)\}}$$

Normalized compression distance (Cilibrasi and Vitányi, 2005: 7; Cilibrasi, 2007)

Here, *C(x)* is the size of the compressed version of *x* using *c*, and *C(xy)* is the size of the compressed version of *x* and *y* concatenated. In essence, NCD measures the maximal additional size needed to compress both objects together compared with compressing one. The denominator normalizes the result to approximate to [0,1], where 0 means the objects are identical (compressing both together has the same cost as compressing one) and 1 means the objects are completely dissimilar (compressing both together has the same cost as compressing each one individually). The effectiveness of NCD depends on the power of the compression function *c*, and in particular its ability to exploit 'similarities' in the objects which are not explicitly visible. But 'off-the-shelf' compressors such as bzip2[6] are very effective at this, even with completely arbitrary data objects.

Given a set of *n* data objects, CompLearn first computes a distance matrix, recording the NCD between each pair of objects. From this, CompLearn creates an unordered tree representing clustering relationships implicit in the distance matrix. An example of an unordered tree is shown in Figure 3[7] below. In this tree, each data object is represented by a leaf node, and the tree

of the overall informativeness of a text. We are, however, not aware of any previous application of a compression-based approach to the clustering of inflectional classes.

[6] http://www.bzip.org

[7] The node styling in figure 2 is a manual addition, as discussed in section 4.2 below.

structure is designed to correlate the distance between data objects in the tree (that is, the number of tree edges between them) with their NCD distance. Thus data objects close together in the tree are similar, while those far apart are dissimilar[8].

Constructing such a tree from the distance matrix is a challenging computational task. In CompLearn, the structure of the tree is topologically constrained to comprise *n* leaf nodes (corresponding to the data objects) and *n-2* internal nodes, each of order 3. Finding a tree with this structure which is the best fit for the distance matrix is an NP-Hard problem (Cilibrasi 2007, p49), so a best approximation to the optimal tree is constructed using a hill-climbing simulated annealing heuristic approach. Initially an arbitrary tree (meeting the topological constraints) is constructed with the *n* data objects as leaves. Then constraint-preserving modifications to the tree's internal structure are applied randomly, in accordance with a probability distribution which favours frequent small-scale changes to tree structure, with occasional larger-scale reorganisations to avoid getting stuck in local maxima. Each new tree is scored according to how well it pairs up similar data objects and separates dissimilar data objects and on each iteration the best-scoring tree generated so far is retained. The process stops when either the best possible score is attained, or there is no further improvement after a large number (circa 100000) of attempted modifications.

Cilibrasi shows that this procedure produces trees which are good approximations of the relations expressed in the distance matrix. However, as the method has a random probabilistic element, multiple runs on the same data may deliver different results. So it is important to execute multiple runs to check that any solution found is stable (and even then, it may not be the only stable solution).

2.2 Extracting classes from unordered trees

The unordered tree structure returned by CompLearn represents relatedness in the data set, but does not directly generate 'classes'. Indeed every internal node in the tree in figure 3 can be interpreted as a valid partition of the leaves into three clusters of 'related' leaf nodes (the clusters being the leaves reachable from each of the three edges leaving the node), and similarly every edge divides the set of leaves into two clusters. The tree structure itself does not tell us which clusters to choose, it just constrains the set of possible (or sensible) clusters – clusters that respect the relatedness structure of the tree and do not, for example, pick out odd leaves from disparate segments in the tree.

[8] The tree-drawing algorithm used to draw this tree is 'neato' in the Graphviz package (http://www.graphviz.org). This applies its own heuristics to lay out the tree so that nodes close together in the tree are generally also grouped together. This means that it is reasonably safe to interpret the visual clustering of the tree as correlating broadly to tree distance which in turn correlates broadly to distance in the NCD matrix.

In order to derive sensible classes from the tree we start off with a simple assumption: that no single class contains more than half the leaves. This assumption only works if we have some idea of what classes we expect to find, and can control the input data set sufficiently to achieve it – in the current context we can do this fairly easily. As soon as we make this assumption, we can impose order on the tree, by identifying an internal node that splits the tree into clusters, none of which contains more than half the leaves, and nominating it as the root of an ordered tree (there will be at most two such nodes in the tree, and we can pick either one). Once the tree is ordered in this way, its structure provides a natural hierarchy of clusters that respect the relatedness structure of the original unordered tree.

The task of finding a set of classes in such a tree becomes 'find a set of internal nodes in the tree which form a disjoint cover of the leaves (that is, which together dominate all the leaves with no overlaps)'. To do this, we need to know (a) how many classes we think there are, (b) how to identify candidate class sets in the tree of that size and (c) how to decide between competing possible class sets. Once again we have to appeal to our intuitions about the problem to decide how many classes to look for, but we can explore solutions for nearby cases as well. We identify candidate class sets by moving down the tree from the root, successively breaking classes into smaller parts represented by their child nodes until we have at least the requested number of classes.[9]

Our approach to choosing between class sets makes use of a function which generates a score for each class in the set. We choose the set for which the variance of these scores is smallest, that is, the set in which the classes are closest to having the same score. We have experimented with three such class measurement functions:

- **count:** this function simply counts the number of leaves in each class. Hence the best class set is the one in which the classes are closest to being the same size as each other.
- **max:** this function returns the maximum NCD score between leaves in the class. The best class set is one which distributes outliers between the classes, without much regard for the distribution of other leaves between the classes.
- **avg:** this function returns the average NCD score between leaves in the class. The best class set for is one where all the classes capture about the same amount of difference among their leaves (visually, they are about the same size, but unlike **count**, they may be different densities).

[9] The ordered tree is binary except for its root node, which is ternary. So in most cases a class is split into two parts. As a special case we allow the root node to represent two classes, one containing two subtrees the other one (in all possible ways), to avoid overcommitting to the initial three-way distribution of classes.

2.3 Evaluating inflectional class results

In order to assess the success of our approach, we need a way of evaluating the inflectional classes returned by the machine learning method. We achieve this by comparing the returned classes with a predefined ‘right answer’ based on our theoretical intuitions. We have experimented with three ways of representing the ‘right answer’:

- **gold standard:** we simply stipulate what the correct class for each data object is, based on our theoretical intuitions. This is a reasonable objective measure of how well the classification algorithm meets our theoretical expectations.
- **classified gold standard:** we create a data set in which each data object is represented just by its gold standard answer (so for example, the noun *strana* is represented simply by the string ‘classII’) and run the classification algorithm over this set. The result aims to represent the best possible classification that can be achieved using the classification algorithm (without any noise in the input), so that comparison with this set is a good subjective measure of how well the algorithm is coping with the additional noise in ‘real’ data inputs. However, the data objects are very small, so the compression algorithm may not distinguish between them very well.
- **classified exemplars:** we create a data set as in the previous case, but this time each data object is represented by an exemplar data object of the right class (the same exemplar for all objects in one class). As above, classifying this set aims to represent the best possible classification, but by using a richer input representation the compression function may be more effective at calculating NCD scores.

Each of these alternative ‘right answers’ results in a classification for the input data objects. Each experimental run results in another classification for the data objects. In order to evaluate an experiment, we create a mapping between classes in the experimental result and classes in the right answer, and then count how many data objects respect this mapping – that is, how many of them occur in the right answer class that the mapping predicts for them. There are many ways to construct such a mapping between the classifications, and we choose a mapping which maximises the agreement score.

2.5 Summary of the experimental method

In summary, the basic experimental method we use is as follows:

1. Prepare a data set as a set of files, one for each data object;
2. Create the NCD distance matrix from the data set;
3. Create an unordered tree using the probabilistic simulated annealing method (repeating several times to assess stability);

4. Order the tree by identifying a root node, and determine the best classification using one of the three scoring functions (count, max or avg);
5. Evaluate the classification against one of the 'right answer' classifications (gold standard, classified gold standard, or classified exemplars).

3. Experimental data

3.1 Data format

In order to apply this methodology to the learning of inflectional classes, we use noun paradigm table listings as the data objects. An example of such a listing, for the noun *strana* (country), is given in (1). [10]

```
(1) mor sg nom = stran ^ a @".
    mor sg acc = stran ^ u @".
    mor sg gen = stran ^ i @".
    mor sg dat = stran ^ e @".
    mor sg inst = stran ^ o @" ^ j ( u ).
    mor sg prep = stran ^ e @".
    mor sg prep loc = stran ^ e @".
    mor pl nom = stran ^ i.
    mor pl acc = stran ^ i.
    mor pl gen = stran.
    mor pl dat = stran ^ a ^ m.
    mor pl inst = stran ^ a ^ m'i.
    mor pl prep = stran ^ a ^ x.
```

These listings include morphological feature information and the forms themselves in phonological transcription. The caret (^) marks concatenation and the symbol combination @" marks stress.

Such a listing is represented in a plain text file, and the algorithm described above is run over a set of such files. Thus the compression function is applied to such listings individually, and concatenated together in pairs, in order to compute NCD scores. We briefly note a number of features of this representation which may have some bearing on the performance of the algorithm:

1. The list of forms is always presented in the same order in each file. We have not yet explored whether mixing up the order has any bearing on the results.

[10] The `prep` attribute is used in this dataset to represent the prepositional case (i.e. PREP SG and PREP PL in figure 1). This is also called the locative in many descriptions. The combination `mor sg prep loc` is used to represent the 'second locative' . The noun in this case does not really have a separate second locative as the form is the same as for the standard prepositional (or locative). This is discussed in detail by (Brown 2007).

2. We assume that systematic variation of the morphological terms (‘sg’, ‘pl’, ‘nom’ etc.) will not have a significant impact on the results, as the compression algorithm detects the patterns rather than the content.
3. The inclusion of some morphological segmentation information (ie the use of the caret for concatenation) means that the data incorporates some assumptions about morphological structure. However this structure in itself does not determine morphological classes, which is the main focus of our interest. Nevertheless it would be interesting in future to compare our results with using completely unsegmented surface forms.
4. The inclusion of individual noun stems probably does have a significant bearing on the results, as without them many of the listings would be almost identical. However we think that removing stem information would make the learning task too unrealistic to be of interest.
5. The inclusion of stress markers may well have an impact on performance, as stress patterns cut across morphological classes. Stress is not the main focus of the present paper, but we make some observations about it in section 4.

3.2 Data sets

The data set for our experiments are full paradigm listings (as described above) of the most frequent 80 nouns from Zasorina’s (1977) frequency dictionary of Russian. They were generated from a Network Morphology theory representing the first 1500 most frequent noun lexemes implemented in the default-inheritance-based lexical representation language DATR (Evans and Gazdar 1996). Within these 80 nouns, we can distinguish five classes – the four theoretically motivated classes introduced in section 1, plus a small class of irregular nouns classed as ‘other’. Table 1 lists the nouns included in each class:

Class 1	***Class II***	***Class III***	***Class IV***	***Other***
čelovek (person)	armija (army)	cel' (goal)	delo (affair)	leta (summer/ year)
den' (day)	bor'ba (struggle)	čast' (part)	dviženie (movement)	ljudi (people)
dom (house)	doroga (way)	dejatel'nost' (activity)	gosudarstvo (state)	
drug (friend)	forma (form)	dver' (door)	lico (face)	
glaz (eye)	golova (head)	mat' (mother)	mesto (place)	
god (year)	kniga (book)	molodëž' (young people)	obščestvo (society)	
gorod (town)	komnata (room)	mysl' (thought)	okno (window)	
konec (end)	mašina (car)	noč' (night)	otnošenie (relation)	
mir (world)	nauka (science)	oblast' (area)	pis'mo (letter)	

narod (folk)	noga (leg)	pomošč' (help)	proizvodstvo (production)	
otec (father)	partija (party)	poverxnost' (surface)	rastenie (plant)	
raz (occasion)	pravda (truth)	put' (way)	razvitie (development)	
stol (table)	rabota (work)	reč' (speech)	slovo (word)	
svet (light)	ruka (hand)	skorost' (speed)	solnce (sun)	
tovarišč (comrade)	sila (force)	smert' (death)	steklo (glass)	
trud (labour)	storona (side)	step' (steppe)	uslovie (condition)	
vopros (question)	strana (country)	svjaz' (connection)	veščestvo (substance)	
zavod (factory)	voda (water)	vešč' (thing)	xozjajstvo (economy)	
	vojna (war)	vlast' (power)	znakomstvo (acquaintance)	
	zemlja (country)	vozmožnost' (possibility)		
		zhizn (life)		
Size = 18	**Size = 20**	**Size = 21**	**Size = 19**	**Size = 2**

Table 1: Data set (with English glosses) arranged in theoretically motivated ('gold standard') classes.[11]

As discussed in section1, our theoretical model gives us a clear idea of which lines in the paradigm listings correspond to default information and which correspond to principal parts. We remove each type of data independently, so in our experiments, we used three variants of these data sets:

1. Full paradigms, to establish the baseline performance of the method with 'complete' knowledge.
2. Paradigms with default information removed.
3. Paradigms with principal part forms removed.

4. Experimental results

4.1 Validating 'right answer' sets

Our first experiment compared the three alternative versions of the 'right answer' classification, by creating 'classified gold standard' and 'classified exemplar' sets as described above, and classifying them into 5 classes, using each of the three class measurement functions. The evaluating scores for the

[11] The lexemes are given here in transliteration. The actual fragment generates paradigm listings in a lower ASCII phonological transcription.

resulting classifications against the hand-crafted 'gold standard' classification are shown in table 2.

	Class measurement function		
	Count	*Max*	*Avg*
Classified gold standard	77	60	77
Classified exemplar	77	80	80

Table 2: Evaluation scores (out of 80) for classification of 'right answer' data sets against gold standard (number of classes = 5)

These results suggest that the basic classification method performs reasonably well when given 'perfect' data, but that there is a clear benefit to giving it the richer data inputs provided by the exemplar cases. The scores for the 'count' function are interesting, because the algorithm would be trying to find a solution with close to 16 nouns in each class, for which we would expect a much lower score (as at least 14 of the nouns classified as 'other' would be wrong). The fact that the evaluation scores are high suggests that the tree is modeling the relational structure of the data well, and only permits solutions which are close to the correct balance. The relatively low classified gold standard/max score may be indicative of the fact that the data is too simple, so that distances between data objects are all similar and so the 'max' classification is fairly arbitrary.

There results encourage us to focus on the exemplar version of the 'right answer' data, and the 'max' and 'avg' measurement functions, in the remaining experiments.

4.2 Validating full paradigms

In our second experiment we classified the full paradigm data sets and evaluated the results against the true gold standard and the classified exemplar set. The unordered tree resulting from the classification process is shown in figure 3, and the results of the evaluations in table 3.

These results show a consistent level of classification success of about 55/80 (69%) for the real data. It is interesting that the results are the same for all three measurement functions. This may suggest that the constraints captured in the tree structure itself are more significant than different approaches to evaluating classification sets. The leaves in figure 3 are styled to illustrate how the gold standard right answers distribute across the clustering structure of the tree. It is evident that the class II nouns cluster very well, class III fairly well, with a small group of outliers, while classes I and IV are fairly confused (which is perhaps consistent with the Network Morphology account of the close relationship between these classes).

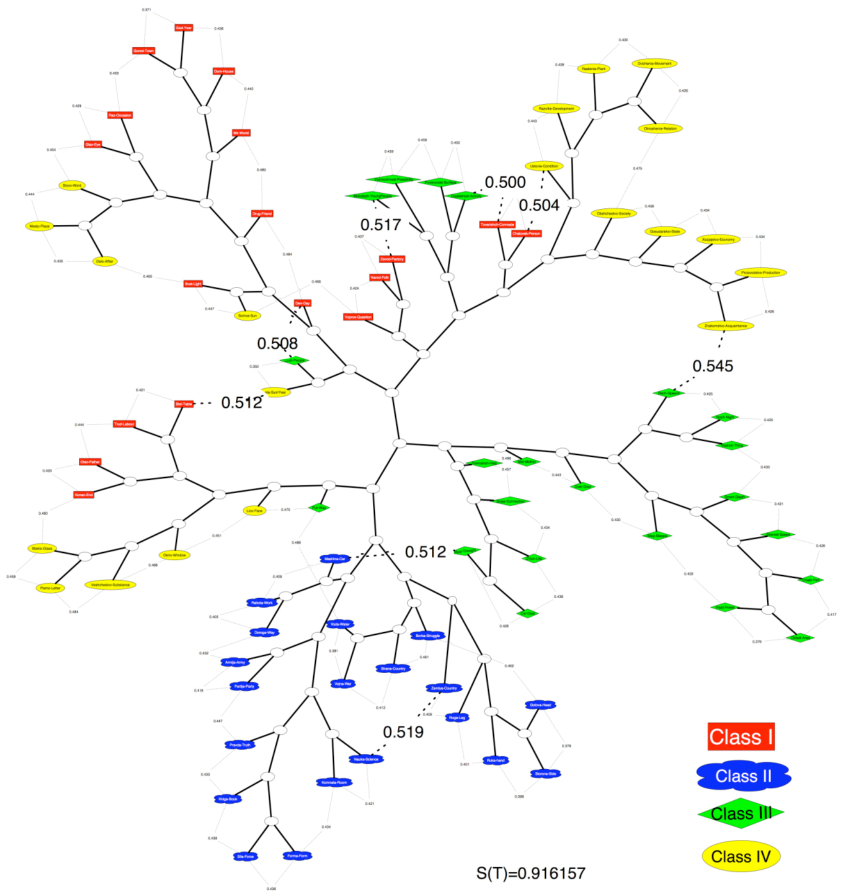

Figure 3: Classification of the full paradigm set – colours/shapes of the leaf nodes correspond to the 'right answer' (gold standard). The tree score (S(T)) is an indication that this tree is considered a good model of the underyling distance matrix.

	Class measurement function		
	Count	***Max***	***Avg***
Gold standard	55	55	55
Classified exemplar	56	55	55

Table 3: Evaluation scores (out of 80) for classification of full paradigm data sets against two 'right answer' representations (number of classes = 5)

4.3 Removing defaults

In our third experiment, we removed single lines associated with default value specifications systematically from all the paradigm listings, reclassified the data and evaluated the results against the gold standard.[12] The results are shown in table 4.

Form removed	**Class measurement function**		
	Count	***Max***	***Avg***
(none)	*55*	*55*	*55*
PREP PL	54	55	55
DAT PL	54	55	52
ACC PL	54	55	50
INS PL	54	55	50
PREP SG	54	55	50
NOM PL	37	35	39

Table 4: Evaluation scores (out of 80) for classification of paradigm data sets with individual default values removed evaluated against the gold standard (number of classes = 5)

This table shows that removal of information provided by default in general makes very little difference to the performance of the classifier. The one exception is the nominative plural case, discussed further in section 5 below.

4.4 Removing principal parts

In our last experiment, we remove single lines associated with principal parts, and so considered essential identifiers of the inflectional class. Results of the evaluation against the gold standard are given in table 5.

[12] Results against the classified exemplar set were the same for the 'max' and 'avg' measures. For 'count' they varied slightly, but not systematically.

Form removed	Class measurement function		
	Count	*Max*	*Avg*
(none)	*55*	*55*	*55*
GEN SG	54	55	61
NOM SG	53	42	50
GEN PL	54	46	46
ACC SG	42	43	43
DAT SG	42	38	39
INS SG	41	38	38

Table 5: Evaluation scores (out of 80) for classification of paradigm data sets with individual principal part values removed evaluated against the gold standard (number of classes = 5)

Here we see much greater variation in the impact of the removal of the data on the classification performance, consistent with the claim that these values are more significant to correct classification. We also see some, but not all, case show a significant variation in performance between measurement functions, which may be an indication of a difference in outlier distribution.

5. Discussion

5.1 Analysis

The results in section 4 indicate that there is little effect on classification when more default-like cells are removed. In contrast, a greater effect appears to be observable when principal-parts-like cells are removed. For example, removal of the oblique plural forms (dative plural, instrumental plural and prepositional plural) has a minimal effect on the correct classification in comparison with the base case, which reflects the fact that these are defaults for all nouns. In contrast the instrumental singular is clearly a good class identifier, as removing it from the paradigm tables has the most significant effect on classification performance.

There are, however, two instances where the effect is not as expected. When the genitive singular is removed a classification score of 61 is achieved relative to the gold standard using the ‘avg’ measurement function. This compares with 55 for the base set, indicating that classification seems to be improved when the genitive singular is absent. More subtly, this effect is not observable when the ‘max’ function is used. This suggests that the genitive singular may contribute to greater variation from average similarity within classes, possibly attributable to the fact that there are essentially two allomorphs shared across the four classes (see Figure 1). Interestingly, if there were no superclass N_O, this particular paradigm cell would be a violation of Carstairs-McCarthy’s (1994) No Blur principle, which essentially requires that a realization is either a default or a class identifier. The second case is the removal of the nominative plural, which has a greater effect than we might expect for a default-like cell. We conjecture that this could be connected with the fact that the inclusion of stress patterns in the dataset give it a greater role in identifying classes than just the affixal morphology would indicate.

5.2 Conclusions

We have presented data from an empirical investigation of defaults and principal parts where we determine the role they play in grouping high frequency nouns by removing the different elements individually and systematically. The experiments so far indicate that there is potentially an observable effect. Removal of default-like information typically has less of an effect than removal of principal parts information.

We have used a naturally occurring data set (the 80 most frequent noun lexemes) to avoid idealizing the task too much. These nouns include a range of complications and irregularities not shown in figure 1, but nevertheless we are able to show some interesting effects. In addition, our data includes stress information which complicates the classification task, because stress patterns cross-classify the nouns in ways which are not straightforwardly predictable from inflectional class and cannot be accounted for purely in phonological terms (see Brown *et al.* 1996). In ongoing work we are checking the degree to which our current results for principal parts and defaults are dependent on data format and exploring the impact of the stress information on the classification task.

5.3 Future work

Our experiments indicate that this approach has significant potential for investigating the role of morphological complexity of the type we have defined earlier. There are a number of core areas which our future work will concentrate on. Further investigation needs to be carried out on the methodology in terms of its stability and evaluation of the clustering. We will also compare our results with the static principal parts analyses which can be created with the online tool referred to in Finkel and Stump (2007). In particular, we can compare the Finkel and Stump scores with the results obtained for our clusterings when the principal parts information is removed. We will also investigate the role of stress in the Russian system and carry out a controlled comparison of the stress patterns and their interaction with inflectional classes. As we can generate the paradigm sets from the underlying theory we can also alter that to eliminate segmentation information and determine its role in classifying inflectional classes.

Acknowledgements

We would like to thank Farrell Ackerman, Matthew Baerman, Anja Belz, Olivier Bonami, Gilles Boyé, Patricia Cabredo Hofherr, Lynne Cahill, Scott Collier, Grev Corbett, Rob Malouf, Paul Marriott, Erich Round, Greg Stump, and participants at the HPSG workshop on morphology and formal grammar for helpful discussion of the ideas we present here. This paper is part of an ongoing project. Special thanks are due to Gerald Gazdar for advice on a variety of things throughout this research. An earlier version was presented at the 'Morphological complexity' workshop, Harvard University, January 2010. The support of the European Research Council in funding Brown's research (grant number ERC-2008-AdG-230268 MORPHOLOGY) is gratefully acknowledged. Any errors are ours.

References

Ackerman, Farrell, Blevins, James P. and Malouf, Rob. 2009. Parts and wholes: implicative patterns in inflectional paradigms. In James P. Blevins and Juliette Blevins (eds.), *Analogy in Grammar: Form and Acquisition*, pages 54-82, Oxford: Oxford University Press.

Aronoff, Mark. 1994. *Morphology by itself: stems and inflectional classes*. Cambridge, Mass.: The M.I.T. Press.

Baerman, Matthew, Brown, Dunstan and Corbett, Greville G. 2005. *The syntax-morphology interface: a study of syncretism*. Cambridge: Cambridge University Press.

Baerman, Matthew, Brown, Dunstan and Corbett, Greville G. 2009. Morphological complexity: a typological perspective. Paper presented at the European Science Foundation Workshop 'Words in Action', Istituto di Linguistica Computazionale "Antonio Zampolli" CNR Pisa, 12-13 October 2009.

Brown, Dunstan. 2007. Peripheral functions and overdifferentiation: The Russian second locative. *Russian Linguistics* 31, 61-76.

Brown, Dunstan, Corbett, Greville G., Fraser, Norman, Hippisley, Andrew and Timberlake, Alan. 1996. Russian noun stress and Network Morphology. *Linguistics* 34, 53-107.

Brown, Dunstan and Hippisley, Andrew. Forthcoming. *Network Morphology*. Cambridge: Cambridge University Press.

Carstairs-McCarthy, Andrew. 1994. Inflection classes, gender, and the Principle of Contrast. *Language* 70, 737-88.

Cilibrasi, Rudi and Vitányi, Paul M. 2005. Clustering by compression. *IEEE Transactions on Information Theory* 51, 1523-1545.

Cilibrasi, Rudi. 2007. *Statistical inference through data compression*. Ph.D. Institute for Logic, Language and Computation, University of Amsterdam.

Corbett, Greville G. and Fraser, Norman. 1993. Network Morphology: a DATR account of Russian inflectional morphology. *Journal of Linguistics* 29, 113-42.

Evans, Roger and Gazdar, Gerald. 1996. DATR: a language for lexical knowledge representation. *Computational Linguistics* 22, 167-216.

Finkel, Raphael and Stump, Gregory T. 2007. Principal parts and morphological typology. *Morphology* 17, 39–75.

Goldsmith, John and O'Brien, Jeremy. 2006. Learning Inflectional Classes. *Language Learning and Development* 2, 219 - 50.

Juola, Patrick. 1998. Measuring linguistic complexity: the morphological tier. *Journal of Quantitative Linguistics* 5, 206–213.

Kettunen, Kimmo, Sadeniemi, Markus, Lindh-Knuutila, Tiina and Honkela, Timo. 2006. Analysis of EU Languages Through Text Compression. In Tapio Salakoski, Filip Ginter, Sampo Pyysalo and Tapio Pahikkala (eds.), *Advances in natural language processing. 5th International Conference on NLP, FinTAL 2006 Turku, Finland, August 23-25, 2006 Proceedings*, pages 99-109. Berlin/Heidelberg: Springer-Verlag.

Stump, Gregory T. and Finkel, Raphael. 2010. Predictability, predictiveness and paradigm complexity. Paper presented at the workshop 'Morphological complexity: implications for the theory of language', Harvard University, January 22, 2010.

Zasorina, L. N. 1977. *Častotnyj slovar' russkogo jazyka*. Moscow: Russkij jazyk.