UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

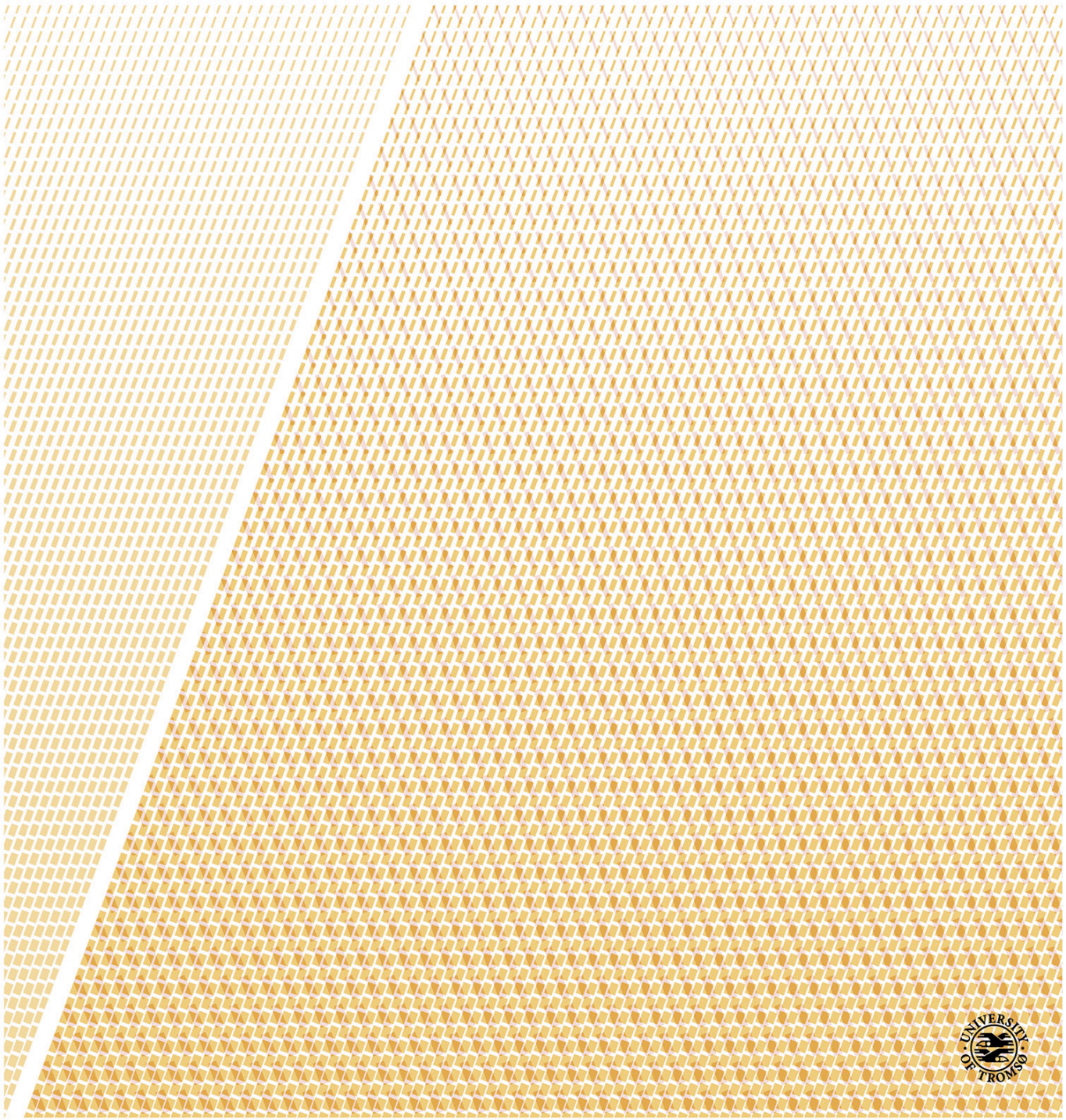
Faculty of Humanities, Social Sciences, and Education

Russian natural language processing for computer-assisted language learning

Capturing the benefits of deep morphological analysis in real-life applications —

Robert J Reynolds

A dissertation for the degree of Philosophiae Doctor – February 2016



Russian natural language processing for computer-assisted language learning

Capturing the benefits of deep morphological analysis in real-life applications

Robert J. Reynolds

A dissertation presented for the degree of
Philosophiae Doctor (PhD)

Faculty of Humanities, Social Sciences, and Education
UiT: The Arctic University of Norway
Norway
February 4, 2016

Printed by Tromsprodukt AS, Tromsø, Norway
ISSN 0000-0000 ISBN 000-00-0000000-0

To Rachael

Contents

List of Figures

List of Tables

Abstract

In this dissertation, I investigate practical and theoretical issues surrounding the use of natural language processing technology in the context of Russian Computer-Assisted Language-Learning, with particular emphasis on morphological analysis.

In Part I, I present linguistic and practical issues surrounding the development and evaluation of two foundational technologies: a two-level morphological analyzer, and a constraint grammar to contextually disambiguate homonymy in the analyzer's output. The analyzer was specially designed for L2 learner applications—with stress annotation and rule-based morphosyntactic disambiguation—and it is competitive with state-of-the-art Russian analyzers. The constraint grammar is designed to have high recall, allowing an L2-learner application to base decisions on all possible readings, and not just the single most likely reading. The constraint grammar resolves 44% of the ambiguity output by the morphological analyzer. A voting setup combining the constraint grammar with a trigram hidden markov model tagger demonstrates how a high-recall grammar can boost performance of probabilistic taggers, which are better suited to capturing highly idiosyncratic facts about collocational tendencies.

In Part II, I present linguistic, theoretical, practical issues surrounding the application of the morphological analyzer and constraint grammar to three real-life computer-assisted language-learning tasks: automatic stress annotation, automatic grammar exercise generation from authentic texts, and automatic evaluation of text readability. The automatic stress placement task is vital for Russian language-learning applications. The morphological analyzer and constraint grammar yield state-of-the-art results, resolving 42% of stress ambiguity in a corpus of running text.

In order to demonstrate the value of a high-recall constraint grammar, I developed Russian grammar activities for the VIEW platform, a system for providing automatic Visual Input Enhancement of Web documents. This system allows teachers and learners to automatically generate grammatical highlighting, identification activities, multiple-choice activities, and fill-in-the-blank activities, enabling them to study grammar using texts that are interesting or relevant to them. I show

that the morphological analysis described above is instrumental not only for generating exercises, but also for providing adaptive feedback, a feature which typically requires encoding specific learner language features.

A final test-case for morphological analysis in Russian language-learning is automatic readability assessment, which can help learners and teachers find texts at appropriate reading levels. I show that features based on morphology are among the most informative for this task.

Acknowledgements

As I finish writing my dissertation, it is poignantly clear that none of this would have been possible without the influence, support, and friendship of so many. Graduate school has been a wonderful experience, but there were several bumps in the road—including some significant health challenges—but at every step, I was blessed to have good people at my side, providing the support that I needed. It is easy to see Norway through rose-colored glasses, having been surrounded by such wonderful people for the last few years.

First and foremost, I am grateful to those who advised my research at UiT. Laura Janda has frequently gone above and beyond the call of duty, not only to help me perform my research, but also to help me and my family become situated in a foreign country. Even though this dissertation is outside of her usual research domain, she learned along with me, and her feedback and direction have been invaluable. I will always appreciate the encouragement and generosity that she showed me throughout this process.

Detmar Meurers' presence at UiT was a wholly unexpected blessing that has made this dissertation much better than it would have been. In 2009, when I arrived at The Ohio State University to begin my Russian Linguistics Master's program, I was disappointed to learn that he had returned home to the University of Tübingen the year before. Years later, after arriving in Tromsø to begin my PhD studies, I was elated to learn that, by some cosmic coincidence, he had just been appointed as a Professor II at UiT. Although he was physically in Tromsø only two weeks each year, he was generous enough to videoconference with me on a regular basis. His expertise was instrumental at every level. I am glad that I caught up with him this time.

Trond Trosterud was an excellent tutor, especially during the early stages of developing the morphological analyzer and constraint grammar. I knew nothing about the computer languages involved, and so he really had to start with me from square one. Because his office door is less than ten meters away from mine, I am afraid that he also fielded the lion's share of my random adminstrative questions. I am indebted to him for his insightful comments at various stages of my research,

and for thoughtful kindness shown to my family on multiple occassions.

While working on my dissertation, I had the privilege of being a member of both the CLEAR[1] research group and the Giellatekno research group at the University of Tromsø. Without exception, the members of these research groups are knowledgable, professional, and warmhearted. It is an honor to be associated with them.

CLEAR is jointly led by Tore Nesset and Laura Janda, with members Aleksandrs Berdicevskis, Hanne Eckhoff, Anna Endresen, Anastasia Makarova, Maria Nordrum, Svetlana Sokolova, Francis Tyers, Julia Kuznetsova, and Olga Lyashevskaya. All of them have given valuable feedback on earlier versions of the research reported in the dissertation, and individually they have been an immense help to me in other ways. Tore Nesset, although not an official advisor, has given excellent feedback and counsel on many parts of this dissertation, including stimulating discussion regarding Russian word stress. Sasha Berdicevskis shared a preprocessed version of the SynTagRus corpus with me, along with several useful scripts. Julia Kuznetsova was kind to share a copy the Exploring Emptiness database of aspectual pairs with me. Anya, Nastya, Sasha and Sveta were each very willing native informants while I was developing the morphological analyzer. Nastya went the extra mile, helping me place stress on hundreds and hundreds of proper nouns. Olga Lyashevskaya kindly directed me to the most recent electronic version of the *Grammatical dictionary of Russian* that Elena Grishina and Andrej Zaliznjak were gracious enough to make freely available for academic purposes. Last, and certainly not least, I owe a special debt of gratitude to Francis Tyers, who was much more than just an excellent office mate and research collaborator. Because of his influence, I have developed many very useful technical skills along the way: LaTeX document markup, bash scripting, meme literacy, vi editing, and subversion, to name a few.

In the Giellatekno research group, I am especially grateful to Lene Antonsen for perceptive feedback, lively discussion, and driving lessons. Heli Uibo deserves special gratitude for her programming and server support for the earliest prototypes of rusVIEW. Sjur Moshagen has generously helped with technical aspects of the morphological analyzer, including design considerations, regression testing, and debugging two-level rules. I am also grateful to Linda Wiechetek—who was my office mate for a short time when I first arrived—for helping orient me to life in Tromsø, especially for pointing out the best hiking trails and fishing spots. I also want to express gratitude to Ciprian Gerstenberger, Børre Gaup, and others who provided indispensable technical support at various times.

Speaking of technical support, I am also indebted to several friends at the

[1] CLEAR stands for Cognitive Linguistics: Empirical Approaches to Russian.

#apertium and #hfst channels on freenode IRC chat, especially Tino Didriksen, Kevin Brubeck Unhammer, and Jonathan Washington, who were always willing to share their expertise with using hfst and vislcg3 in real applications. I also want to thank Eduard Schaf for his java programming contributions to rusVIEW. Without him, many of the ideas implemented in Chapter 5 would still just be ideas.

None of this research would not have been possible without the funding for my position from the Faculty of Humanities, Social Sciences and Education. In addition to funding my position, the Faculty also awarded me a grant to pay for programmers, and to purchase materials for the readability corpora used in Chapter 6.

My journey to writing this dissertation began long before I entered graduate school. I am grateful to Mrs. Burraston and Ms. Buckner, who encouraged my study of Russian before I had even reached High School. While living in Russia from 2000 to 2002, I had the privilege of becoming friends with Natasha Tsaryov, Sasha Tsaryov, Vova Blinkov, Baba Nina, Stepan Vasiljevich, Sergey, Shakir, Lilit, the Ilaryonov family, and many others. Thank you for making Russia feel like a second home. After I returned home and began teaching Russian, I was inspired by many coworkers, not least of which are Ken Packer, Chris Porter, Russ Sivertsen, Brent Dance, Rich Hoopes, Tommy Jones, Rob Stephenson, Inna Danilyan, Jake Rees, Sasha Brattos, Arina Purcella, Devin Anderson, Chris Storey, Jacob Burdis, Angela Ellsworth, Julia Carlson, and many others. I learned a lot from each of you. Thanks for all the good memories!

I am very grateful for all of the excellent Russian professors at Brigham Young University, where I received my Bachelor's degree. Thank you to David Hart and Tony Brown, who encouraged me to pursue graduate degrees in Russian. Together with Grant Lundberg, Raissa Vulfovna Solovieva, Michael Kelly, and Jennifer Bown, they are, in my mind, the All-American first team of college Russian instructors.

I began my graduate studies at The Ohio State University, and I owe my professors and colleagues there a debt of gratitude. To my Master's advisor, Andrea Sims, thank you for being a rigorous scholar and dedicated teacher, and for having such high expectations for me. I especially want to thank Jeff Parker for being a professional and capable colleague, and the best of friends. By moving to Norway, I proved that our families are, in fact, distinct entities, but I certainly look forward to problematizing that issue again in the future. I want to give a big shoutout to Dan Davidson, Mike Furman, Kate White, Lauren Ressue, Mike Phelan, Nina Haviernikova, Yuliia Aloshycheva, Spencer Robinson, Dusty Wilmes, Monica Vickers, and all the other graduate students in the Slavic and Linguistics departments at OSU that helped me grow as a scholar and made my time there so memorable. Finally, thank you to Brian Joseph and the Distinctive Features, who

taught me that there is more to graduate school than winning.

Moving a young family to a foreign country was a daunting task, and I want to thank my children for meeting the challenge with courage and excitement. I am proud of them for how quickly they adapted to living in a new place, learning a new language, and making new friends. Last, and most of all, I want to thank my wife, Rachael, for sharing this adventure with me. She has been my support through thick and thin, and she has sacrificed many comforts over the years for us to chase this dream. I never could have done this without her.

Preface

Although the structure and formatting of this dissertation conform to established norms in linguistics and computer science, there is one notable exception. The first has to do with the transliteration—or romanization—of cyrillic characters. The customary transliteration system in linguistics scholarship is the aptly named *scholarly* or *scientific* transliteration scheme, which I follow throughout the dissertation, except in Chapter 2, which discusses at length many rules in the two-level morphology formalism. This formalism, as an instance of finite-state modeling, is inherently concerned with one-to-one mappings of characters, but the scholarly transliteration system incorporates several correspondences in which cyrillic characters are represented as digraphs in the latin alphabet. This makes visual representations of fixed-width alignments of one-to-one mappings cumbersome, if not completely illegible. Therefore, all discussion of two-level rules, which is limited to Chapter 2, makes use of the ISO9 transliteration system which exhibits a strict one-to-one mapping of characters. Both systems are given in Table 1.

	а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н	о	п
Scholarly	a	b	v	g	d	e	ë	ž	z	i	j	k	l	m	n	o	p
ISO9	a	b	v	g	d	e	ë	ž	z	i	j	k	l	m	n	o	p

	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
Scholarly	r	s	t	u	f	x	c	č	š	šč	"	y	'	è	ju	ja
ISO9	r	s	t	u	f	h	c	č	š	ŝ	"	y	'	è	û	â

Table 1: Comparison of Scholarly and ISO9 transliteration systems

Although ISO9 may seem foreign to trained Slavicists, the differences between the two systems are not extensive, and the diacritics used in ISO9 are intuitive enough that reading should not be difficult.

Authorship Because of the broad scope of the dissertation, some of the research reported herein was completed in collaboration with researchers with specialization in relevant disciplines, or with required programming skills. Such cases are

indicated by a footnote at the beginning of the chapter. For those chapters reporting research to which others contributed, I primarily use first-person plural pronouns. However, there are some specific contexts in which the plural would have been distracting, cumbersome or confusing. For example, when summarizing chapters consecutively, I use singular pronouns throughout, even though the research reported in one or more of the chapters was completed in cooperation with others. The use of singular pronouns is in no way meant to diminish the contributions of my collaborators.

Chapter independence One other result of having a broad scope is that different chapters in the dissertation will be interesting to different audiences. For this reason, each chapter is written to stand alone, to a certain degree, without necessarily requiring the reader to be familiar with preceding chapters. This means that sometimes information is repeated in more than one chapter. For those who read the chapters sequentially, this may have the unintended effect of making the dissertation more memorable.

Chapter 1

Introduction

1.1 Introduction

This dissertation focuses on the linguistic and computational analysis of Russian in the context of language learning. Among major world languages, Russian has relatively rich morphology—both derivational and inflectional—and experienced Russian instructors consider morphological complexity to be the most prominent source of difficulty for Russian second language learners (Leaver et al., 2004, p. 126–127).

To address this difficulty, computer-assisted language learning tools can be used to deliver mechanical drills, asking learners to supply a particular morphological form when given the base form. However, more and more empirical studies of language acquisition provide evidence that such mechanical drilling exercises are not as effective as communicative focus-on-form activities in which the learners focus on target grammatical structures incidental to a real communicative task (Wong and Van Patten, 2003, 2004, and citations therein). This idea has met resistance among Russian practitioners on the grounds that Russian is a more difficult language, and therefore requires different methods for full acquisition (Leaver et al., 2004). However, to this point, no empirical studies have demonstrated advantages to the traditional drilling approach with Russian. To the contrary, Comer and deBenedette (2011) showed that learners studying a morphologically difficult set of constructions[1] showed more complete acquisition when using focus-on-form exercises. One group conducted mechanical production exercises and another group was given activities that asked students to "interpret the grammatical forms in the input and map those forms to destinational or locational meanings." Learners in

[1]The constructions investigated surrounded the use of prepositional versus accusative case when expressing location versus destination following the prepositions *v* 'in(to)' and *na* 'on(to)', which includes five potential surface inflections.

the group with mechanical drilling did not show significant learning gains in interpreting these constructions, and overgeneralized the destinational construction when producing these constructions. On the other hand, the group combining form and meaning showed significant gains in interpretation and production of both constructions, even though they were not required to produce the form throughout the entire treatment. This is only one study, but it casts doubt on the notion that Russian morphology requires mechanical drilling for acquisition.

Creating focus-on-form exercises for morphological constructions can be time-intensive, which can be a deterrent to instructors. However, research in natural language processing has established a variety of robust approaches to automatic morphological analysis, which opens the possibility of generating such activities automatically.

The intersection of computer-assisted language learning and natural language processing has a relatively short history, gaining real traction only two decades ago (Nerbonne, 2003). The research presented in this dissertation is most relevant for what Meurers (2012) refers to as Authentic Text Intelligent Computer-Assisted Language Learning (ATICALL). ATICALL is primarily concerned with tools for selecting, enhancing, or adapting authentic texts for language learners, as opposed to analyzing learner language, which is an important component of intelligent tutoring systems, automated scoring, and learner corpora processing.

In terms of intended functionality, the technology described herein is intended to support selection and enhancement of Russian authentic texts. As such, these technologies are designed to process authentic native-Russian texts, with linguistic and computational focus on morphology. They are distinguished from other state-of-the-art morphological analyzers by the combination of two capabilities that are crucial for Russian language-learning applications: analysis/generation of stressed wordforms, and high-recall[2] morphosyntactic disambiguation.

ATICALL is inherently interdisciplinary, relying on contributions from computational linguistics, second language acquisition, theoretical linguistics, instructional design, and others. This variety of research methods and topics in the field is reflected in the broad scope which is taken in the dissertation. As described in more detail in Section 1.2 below, I present research on morphological analysis, morphosyntactic disambiguation, automatic word stress annotation, dynamic grammar excercise generation, and automatic L2 readability classification of texts. Although these topics may seem disparate, they are all tied together by their connection to one central theme, which is the provision of language processing tools to support the selection and enhancement of authentic Russian texts. The current chapter serves as a broad introduction to this core idea, leaving more specific in-

[2]The term "high-recall" disambiguation refers to the goal of never removing correct morphosyntactic readings.

troductory remarks to each chapter.

Russian Russian grammar books are dominated by morphology-centered themes, such as inflectional paradigms, case governance, and modifier agreement. Russian's inflectional morphology is predominantly fusional, which means that a single inflectional affix denotes a complex of morphosyntactic values. The typical noun paradigm has twelve cells: six cases, singular and plural, with lexically specified gender. Modifier paradigms have as many as 30 cells, including attributive forms, predicative forms (short-forms), and comparatives. Verbs have four past tense forms, six non-past forms, imperatives, verbal adverbs, and participles, yielding as many as 121 cells (Janda and Lyashevskaya, 2011, p. 719). Because morphology plays such a central role in Russian language learning, this dissertation is focused primarily on technologies that can automatically process and manipulate morphological structures.

Natural language processing methods The choice of natural language processing methods in this dissertation was motivated by the nature of the language-learning applications described herein. Approaches to natural language processing can be divided into two overarching categories. *Rule-based* approaches are built by linguists who formalize linguistic generalizations. On the other hand, *probabilistic* approaches rely on machine-learning of models based on large gold-standard corpora. Probabilistic methods have become popular for a variety of reasons, but they pose some problems for computer-assisted language learning applications. First, the output of a probabilistic model can be unpredictable. Errors are caused by mysterious interactions in the training data, and errors can only be corrected by getting more and/or better data to train on, and even then improved results are not guaranteed. On the other hand, rule-based approaches are rationally deterministic, and errors can be manually corrected. The rational foundation of rule-based approaches allows you to build a system with intuitionistic/epistemic logic, in which you "know whether you know". This is important in computer-assisted language learning applications, because it allows you to avoid tokens which the system cannot draw sure conclusions about. Natural language processing systems are not perfect, and errors have the potential to confuse or even discourage learners. Whereas a probabilistic model blindly gives the most probable output, a rule-based model can be tuned to only give output that is certain, allowing it to fail gracefully.

Another reason to prefer rule-based approaches for the research presented in this dissertation is the need to not only analyze, but also to generate wordforms for a number of grammar exercises. As discussed in Chapter 4, one unique requirement of Russian language learners is explicit annotation of stress position, which is missing from most state-of-the-art Russian morphological engines. In order to

fill this need, I designed a finite-state transducer using the two-level formalism (Koskenniemi, 1983), which allowed for efficient encoding of shifting stress patterns, as described in Chapter 2 below.

Pedagogical foundations The interventions developed in this dissertation are based on modern research in second language acquisition theory. One of the most broadly accepted ideas in second language acquisition is that extensive exposure to meaningful, comprehensible input is essential to successfully acquire a second language (Long, 1981, 1983; Krashen and Terrell, 1983; Swain, 1985, 2005; Robinson et al., 2012).

Although Krashen has taken a radical position that *only* input is necessary for acquisition (Krashen, 1977, 1985), other theoreticians have suggested that input is not sufficient. Schmidt (1990, 2010) argued for the *Noticing Hypothesis*, which is the claim that a learner must consciously notice language categories and forms—such as inflectional morphology—in order to acquire those forms. Sharwood Smith (Sharwood Smith, 1981, 1991, 1993) developed the concept of input enhancement[3], which is "the manipulation of selected (usually linguistic) features of the input deemed important by the language teachers or teaching materials creators with the specific aim of speeding up [L2] development." (Sharwood Smith, 2014, p. 38). As suggested by his definition, input enhancement can take many forms, but the type of input enhancement most relevant to this dissertation is *visual* input enhancement, which consists of highlighting parts of a text in order to heighten learners' awareness of a given grammatical feature (Polio, 2007; Sharwood Smith, 2014). Empirical evaluations of the effects of visual input enhancement have had mixed results (Lee and Huang, 2008; Leow, 2009, and citations therein), with positive, neutral, and negative effects reported for both learning grammatical forms and comprehending the target texts. Clearly, more research is needed to understand these divergent results.

The applications discussed in this dissertation are based on the concepts of input enhancement and noticing. Although these concepts can supply a rational theoretical foundation for the ATICALL enterprise, they have not received sufficient empirical support. Computer applications such as those presented in this dissertation have the potential to supply a structured testbed for future research of input enhancement and the noticing hypothesis.

[3] Also called "Consciousness Raising" in the earlier works.

1.2 Structure of the dissertation

The dissertation is divided into two main parts. Part I is devoted to the conceptual issues surrounding the development of foundational natural language processing tools for Russian: morphological analysis and context-based morphosyntactic disambiguation. Part II focuses on higher-level tasks that take advantage of these tools: automatic word stress annotation, automatic generation of grammar exercises within authentic texts, and automatic classification of texts according to second-language reading level.

In Chapter 2, I present a new morphological analyzer, based on the *Grammatical dictionary of Russian* (Zaliznjak, 1977). The analyzer is built using a two-level formalism, where each wordform is given an "underlying" form that is then transformed into an actual surface form by means of 29 rules. By using context-based rules to capture morphophonological and orthographic generalizations, I significantly reduce the complexity of the lexicon. I give a thorough description of many of these rules, and demonstrate that the new morphological analyzer is competitive with existing, free state-of-the-art analyzers.

In Chapter 3, I present work on a Russian constraint grammar, tuned to have high recall, i.e., remove readings conservatively, always avoiding removing correct readings. Russian has widespread homonymy, and the constraint grammar removes readings from tokens in running text based on the surrounding context. The constraint grammar contains 299 rules, which are ranked in groups according to their reliability. I evaluate the grammar against a gold corpus, and give both a quantitative and qualitative breakdown of its performance. I also combine the grammar with a probabilistic trigram tagger and show that the combination outperforms each individual tagger, even with less training data.

In Chapter 4, I present research addressing the automatic word stress annotation task, which is an essential language-learning application of the morphological analyzer and constraint grammar. Russian has a number of complex stress patterns that are assigned lexically. Because word stress is not marked in standard Russian orthography, mastering these complex stress patterns is difficult for language learners. Support for determining word stress position facilitates ICALL activities that can help learners to practice word stress placement with authentic texts. Based on the output of the morphological analyzer and constraint grammar, I evaluate a number of algorithms against a gold corpus of stress-annotated running text, with state-of-the-art results.

Chapter 5 presents a higher-level application of the morphological analyzer and constraint grammar: dynamic generation of grammar exercises in online texts. In this chapter, I describe issues surrounding the development of Russian grammar activities on the VIEW platform, demonstrating the utility of the morphological

analysis tools for such applications. I also demonstrate the possibility of generating adaptive feedback based on this native-language NLP, thereby invalidating the common assumption that adaptive feedback to learner responses is only possible with learner-language NLP.

The last study of the dissertation, presented in Chapter 6, explores the use of my morphological analysis tools in the automatic second-language readability classification task. The ability to automatically identify a Russian text's L2 readability level is a natural complement to the automatic grammar generation described in the previous chapter, since it allows teachers and learners to find appropriate texts more easily. In this chapter, I describe work to collect a gold corpus of Russian L2 readability, as well as building probabilistic classifiers to automatically rate the readability of unseen documents.

Finally, in Chapter 7, I summarize the conclusions of these studies, and outline ways in which future research can build on these results.

Part I

Linguistic analysis and computational linguistic methods

Chapter 2

A new finite-state morphological analyzer of Russian

This chapter describes UDAR, a new Finite-State Transducer Russian morphological analyzer/generator designed for language-learning applications, particularly those that deal with stressed wordforms. UDAR is written in the lexc and twolc languages and can be compiled using xfst or hfst. I give an explanation of the structure of the transducer, including a description of its morphosyntactic tags, lexicon structure, and two-level rules. I also evaluate its performance in comparison with state-of-the-art Russian morphological engines. The chapter concludes with a brief description of potential applications that this technology supports.

2.1 Introduction

The present chapter is a description and evaluation of UDAR,[1] a new Russian morphological analyzer/generator designed specifically for use in free and open-source intelligent computer-assisted language learning applications. Compared to other major world languages, Russian has a relatively extensive morphology, exhibiting both complex fusional inflection, and productive derivational morphology. Many part-of-speech tagging resources exist for Russian.[2] However, almost none of the existing resources fulfill all of the requirements of free and open-source language-learning applications, most notably because they lack the ability to mark stress.

Russian word stress is difficult for a variety of reasons. First, word stress is almost never marked in written Russian, the only common exceptions being texts

[1] UDAR is an abbreviated form of *udarénie* 'word stress', and it is also a recursive acronym: "UDAR Does Accented Russian."

[2] Throughout this dissertation, I use the term part-of-speech tagging to refer to *detailed* part-of-speech tagging, with morphosyntactic tags specifying number, gender, aspect, etc.

for beginning readers and foreign language learners.[3] Second, because Russian has strong vowel reduction, it is impossible to determine a word's pronunciation without first knowing where (or whether) a word is stressed. In this way, Russian is similar to Arabic and Hebrew, since vowel qualities are underspecified in the standard orthography. Third, word stress distinguishes between minimal pairs, and can therefore be seen as being phonemic. Stress can distinguish between wordforms of a single lexeme (e.g. *déla* 'matter.SG-GEN' vs. *delá* 'matter.PL-NOM/ACC'), wordforms of different lexemes with identical morphosyntax (e.g. *zámok* 'castle.SG-NOM' vs. *zamók 'lock.*SG-NOM'), and wordforms of different lexemes with differing morphosyntax (e.g. *doróga* 'road.N-SG-NOM' vs. *dorogá* 'dear.ADJ-SG-FEM-PRED'). Fourth, Russian has complex patterns of shifting stress which cannot be deduced from stem shape. In other words, it is impossible to reliably predict the stress position on unknown wordforms, especially for language learners. Because most existing morphological engines are designed to analyze and process the (unstressed) standard language, they are unsuitable for one of the primary needs of language learners, as well as other applications with relation to phonetic realization and written text, such as text-to-speech, speech recognition, etc.

This chapter has the following structure. Section 2.2 gives an overview of existing Russian part-of-speech taggers, including a brief description of the highly influential *Grammatical dictionary of Russian* (Zaliznjak, 1977), which serves as the basis of virtually all lexicon-based morphological engines of Russian, including UDAR. Section 2.3 gives an overview of the structure of UDAR, including examples of how particular properties of Russian orthography and morphophonology are handled with the two-level formalism. I also give an overview of UDAR's morphophonological tags, and briefly highlight the different 'flavors'—or variants—of the transducer. Section 2.4 compares the speed, coverage, and accuracy of UDAR with available morphological transducers. Section 2.5 describes potential applications of UDAR, including some which have already been implemented. Section 2.6 contains some concluding remarks and some notes about future research with UDAR.

2.2 Background of Russian part-of-speech tagging

Russian is characterized by fusional morphology with relatively extensive inflection. The prototypical noun paradigm has 12 cells (six cases, singular and plural).[4]

[3] Standard Russian does mark stress on words with ambiguous stress—stress that a native reader would be unable to determine from context. However, such circumstances are relatively rare.

[4] Some subsets of nouns have one or more additional forms: 1) an alternative genitive form, used primarily in partitive constructions, 2) a special locative case, used exclusively with the prepositions *v*

The prototypical adjective paradigm has 34 cells, inflecting for seven cases (counting two variants of the accusative case: inanimate and animate), three singular genders and plural, as well as comparative and predicative forms. Verbal morphology includes inflection for gender and number in the past tense, person and number in the nonpast (present or future), imperatives, verbal adverbs, and four types of participle. In addition, transitive imperfective verbs can be inflected for the passive voice using the suffix *-sâ*. Including the adjectival inflection of the participles, many verbs have as many as 121 paradigm cells (Janda and Lyashevskaya, 2011, p. 721).

Approaches to Russian part-of-speech tagging have historically gravitated toward rule- and lexicon-based methods. This approach is greatly facilitated by the existence of Zaliznjak's *Grammatical dictionary of Russian* (Zaliznjak, 1977), a forward-minded dictionary which assigns a set of inflectional codes to more than 100 000 words. For example, the noun *avtomát* 'automaton, sub-machine gun' is assigned the code м 1a, where м indicates masculine inanimate gender and declension class 1; the number 1 indicates a non-palatalized paired consonant stem; and the latin letter a indicates fixed stress on the stem. Another example is the verb *blagodarít'* 'to thank', whose code is нсв 4b. The нсв indicates that the verb is imperfective and transitive.[5] The 4 indicates the so-called -i- conjugation, and the b indicates fixed stress on the ending. These two examples are very straightforward, but many other symbols are used to mark exceptions and collocational idiosyncrasies, where necessary. In this way, Zaliznjak achieved a fine-grained and impressively accurate formal description of Russian morphology, with quite broad coverage.

With such a rich resource at their disposal, computational linguists have been able to make Russian morphological engines, using Zaliznjak's dictionary as a template. In the following paragraphs, I discuss the most prominent Russian part-of-speech taggers that have been described in scientific publications, ignoring those that are not freely available or are proprietary. Almost all of these analyzers are ultimately based on Zaliznjak's dictionary, with varying degress of completeness.

RUSTWOL

One of the earliest approaches to Russian morphology described in the scientific literature was RUSTWOL (Vilkki, 1997, 2005), which is strikingly similar to UDAR,

'in' and *na* 'on', and 3) a vocative form of nouns referring to persons. In addition, a given paradigm cell may have more than one possible wordform. For instance, many nouns have more than one nominative plural or genitive plural, each with particular semantic connotations, e.g. *syny* 'son.PL-NOM(fig.)' vs. *synov'â* 'son.PL-NOM(lit.)'.

[5] Zaliznjak explicitly marks intransitive verbs with нп. Any verb without this code can potentially be transitive.

since it also uses a two-level morphology. However, it does not include stressed wordforms. RUSTWOL was used to annotate the HANCO corpus (Kopotev and Mustajoki, 2003). RUSTWOL is now developed commercially by Lingsoft, and is no longer open-source.

StarLing

StarLing is a DOS/Windows program primarily designed for work in typological linguistics, but it also includes a morphological engine for analyzing and generating Russian wordforms (Krylov and Starostin, 2003).[6] It was first released in 2000. StarLing is free, and it processes stressed wordforms, but it is not open-source, with stable versions only available for the Windows operating system. This makes it unsuitable for use on most web servers, which predominantly use Linux operating systems.

Dialing/AOT

The Dialing Project (1999–2001) aimed to build a Russian-to-English machine translation system (Nozhov, 2003). The results of the project are open-source and freely available,[7] including a morphological analyzer/generator with stressed wordforms. However, Nozhov (2003, p.112) reports that the analyzer processes words at a rate of 200 megabytes per hour, which is too slow for interactive language-learning applications.

It should be noted that the OpenCorpora project,[8] which aims to be a free and open alternative to the Russian National Corpus,[9] took its original lexicon from Dialing/AOT, and then modified and expanded it according to its needs for corpus annotation (Boxarov et al., 2013). The OpenCorpora lexicon does not contain stress markings. The lexicon from OpenCorpora is used for evaluation in Section 2.4 below.

Pymorphy2

Pymorphy2[10] is a Russian transducer built in python, with optional C++ extensions for increased processing speed (Korobov, 2015). Pymorphy2 does not process

[6] StarLing can be downloaded from `http://starling.rinet.ru/downl.php`. Its Russian morphological analysis can be accessed online at `http://starling.rinet.ru/morpho.php?lan=en`.

[7] `http://www.aot.ru`

[8] `http://www.opencorpora.org`

[9] `http://www.ruscorpora.ru`

[10] `http://www.github.com/kmike/pymorphy2`

stressed wordforms, since its lexicon is taken directly from OpenCorpora, which did not retain the stress annotation of AOT.[11] It includes unknown word guessing and frequency-based weighting of readings.

Mystem

Mystem is a Russian morphological analyzer/generator developed by one of the founders of the Russian technology giant Yandex (Segalovich, 2003). It is distributed freely, and although it is not open-source, it is included for discussion here because of its importance for Russian linguistics research. Specifically, mystem was used to annotate most of the Russian National Corpus. It includes unknown word guessing, and the most recent version (mystem3) offers morphosyntactic disambiguation.

Mocky

Recently, work has been done to apply probablistic language models to Russian part-of-speech tagging. Most notably, Sharoff et al. (2008a) developed a positional tagset for Russian, based on the MULTEXT-East specifications (Erjavec, 2004), and converted a portion of the Russian National Corpus to the new tagset. Three probabilistic taggers were then trained on these data: TnT (Brants, 2000), TreeTagger (Schmid, 2004), and SVM Tagger (Giménez and Màrquez, 2004). Although their reported tagging results are very good (>95% with TnT), these models cannot be used in many language-learning applications because they cannot be used to generate wordforms, and they are blind to stress marking. One notable extension of this work is Sharoff and Nivre (2011), which resulted in a freely available language model for syntactic parsing using MaltParser (Hall et al., 2009).[12]

Summary

Table 2.1 provides a simple summary of the Russian morphological analyzers discussed above, as well as the target features of UDAR, which will be discussed below. The columns show the year of first formal publication (a plus sign indicates that the system is still under active development), platforms or operating systems on which the system is designed, whether the system can intelligently analyze or generate stressed wordforms, whether the system is free and open-source, whether

[11] An earlier version of pymorphy (`https://bitbucket.org/kmike/pymorphy/` was based directly on the AOT lexicon, with stress annotation. However, pymorphy1 operated at very low speeds (a few hundred words per second), which is too slow for most interactive applications.

[12] More information about these resources can be found at `http://corpus.leeds.ac.uk/mocky/`

the system can generate wordforms, and whether the system can disambiguate multiple readings of a token based on sentential context.

	year	platform/OS	stress	FOSS	gen.	disamb.
RUSTWOL	1997	unknown	-	-	+	-
StarLing	2003	DOS/Windows	+	-	+	-
DiaLing/AOT	2003	Windows/Linux	+	+	+	-
pymorphy2	2008+	python (any OS)	-	+	+	-
mystem3	2003+	all major OSes	-	-	+	+
mocky	2008	Linux/Win/Mac	-	+	-	+
UDAR	2015+	Linux/Mac/Win	+	+	+	+

Table 2.1: Comparison of existing Russian morphological analyzers. FOSS = free and open-source software; gen. = can generate wordforms; disamb. = can disambiguate wordforms with more than one reading based on sentential context

2.3 UDAR

The resources discussed in the previous section have left a gap in the possibilities for analyzing and generating Russian. None of them provide the possibility of analyzing and generating stressed wordforms in free, open-source language-learning applications, with the possibility of disambiguating a tokens readings based on context. UDAR was designed to fill this gap. In the following sections, I give a brief description of UDAR, including its lexicon structure, phonological/orthographic rules, and morphosyntactic tags.

2.3.1 Lexc and Twolc

This section contains a very simple explanation of finite-state transducers, as well as the two-level formalisms used to write UDAR: the `lexc` language for creating the lexical network of underlying forms and the `twolc` language for realizing orthographic and morphophonological rules on the underlying forms to produce well-formed surface forms.[13] For a more comprehensive introduction to `lexc` and `twolc`, please see Beesley and Karttunen (2003).

The `lexc` and `twolc` source files can be compiled using both Xerox Finite-State Tools (XFST) (Beesley and Karttunen, 2003) and Helsinki Finite-State Trans-

[13]Note that the term *morphophonological* is used here in a loose sense. The engineering question of which phenomena are handled in `lexc` and which in `twolc` may or may not have any direct relation to the traditional categories of linguistic theory.

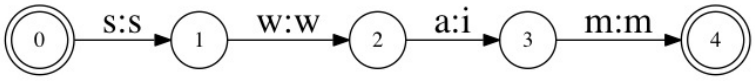


Figure 2.1: Finite-state transducer network

ducer Technology (HFST) (Linden et al., 2011).[14] The lexicon of UDAR is taken primarily from Zaliznjak (1977), by way of a digital copy of the 2001 edition.[15] The 2001 edition of the dictionary includes an appendix of proper names which was also included in the lexicon. I also added all lexemes from Grishina and Lyashevskaya (2008), a list of words extracted from the Russian National Corpus that are not found in Zaliznjak's dictionary. These lexemes are annotated with Zaliznjak-style morphological codes.

Finite-state transducers

A finite-state transducer (FST) is a finite-state automaton in which each transition from state to state consists of an input-output pair. Figure 2.1 shows the behavior of a very simple FST with an English example. This FST converts the string `swam` to its lemma `swim`. The transducer traverses the input string one character at a time. If it encounters the input `s-w-a-m`, then it will output `swim`. Any other sequence of input strings will result in no output.

The transducer in Figure 2.1 can be represented in `lexc` as shown in (1).

(1)
```
swam:swim # ;
```

The input is on the left of the colon, the output is to the right of the colon, the pound sign signals the end of the word, and the semicolon marks the end of the entry. Only a slight modification to this transducer will turn it into a tagger, as in (2):

(2)
```
swam:swim+Pst # ;
```

When the transducer compiled from the `lexc` code in (2) encounters the string `swam`, it will output the lemma-tag pair `swim+Pst`. Additionally, because of their closure properties, finite-state transducers can be reversed, so that our tagger can become a wordform generator. When given the input `swim+Pst`, the reversed

[14] XFST and HFST are both free (i.e. gratis), but HFST is open-source, and has more extensive functionality than XFST. Currently, XFST can be found at http://fsmbook.com and HFST at http://kitwiki.csc.fi/twiki/bin/view/KitWiki/HfstHome.

[15] I am indebted to Andrej Zaliznjak and Elena Grishina for their kindness in making this text available for academic purposes.

FST will output the past tense form `swam`.

In order to create more complex paradigms, `lexc` uses continuation classes, labelled `LEXICON`, to define which continuations—usually endings—belong to which lexemes. If the line of code ends in a `LEXICON` name instead of a #, then that line is continued by every line contained in that `LEXICON`. For example, the `lexc` code given in (3) and (4) result in identical lexical networks.

(3)
```
work+Inf:work # ;
work+Pres+Sg1:work # ;
work+Pres+Sg3:works # ;
work+Past:worked # ;
be+Inf:be # ;
be+Pres+Sg1:am # ;
be+Pres+Sg3:is # ;
be+Past+Sg:was # ;
```

(4)
```
work:work RegularVerb ;
be:  BE ;

LEXICON RegularVerb
+Inf:  # ;
+Pres+Sg1:  # ;
+Pres+Sg3:s # ;
+Past:ed # ;

LEXICON BE
+Inf:be # ;
+Pres+Sg1:am # ;
+Pres+Sg3:is # ;
+Past+Sg:was # ;
```

The obvious advantage of using continuation classes, as shown in (4), is that the full paradigm of new regular verbs can be added with only one line of code, as in (5).

(5)
```
walk:walk RegularVerb ;
talk:talk RegularVerb ;
```

Returning back to Russian, there are times when morphophonological alternations and orthographic rules cause surface-level spelling differences that would require multiplying the number of continuation classes. For example, the nominative plural ending for many nouns is *-y* as in *stol~stoly* ‘table~tables’. However, stems

endings in a velar (*g*, *k* or *x*) cannot be followed by *-y*, but rather have the ending *-i*, as in *mal'čik~mal'čiki* 'boy~boys'. Rather than write a new continuation class in `lexc`, one can generate ill-formed surface forms that can be corrected by a second layer representing orthographic and morphophonological rules of the language. In this case, we can use `lexc` to generate the form *mal'čiky*, and then write a rule that changes all endings in *-ky* to *-ki*. This possibility was made computationally feasible by a ingenious approach developed by Koskenniemi (1983), commonly referred to as a 'two-level morphology.' The `twolc` language is an implementation of a two-level morphology.

Encoding a language's morphology as a finite-state machine has several benefits. Finite-state machines are mathematically elegant, and algorithms have been developed to efficiently combine and modify them. They are computationally efficient, capable of being minimized with extremely high compression rates. A finite-state machine can scale up to hundreds of thousands of states and arcs, allowing for models with millions of words. Furthermore, the lookup process is language-independent, which means that the lexical and grammatical facts of Russian are kept distinct from the language-independent functions of the morphological engine. Another way of saying this is that the source files for a given language are declarative.

Finite-state machines have been used to perform tokenization, syllabification, morphological analysis, spell-checking, word-to-number mapping, lookup words in simple dictionaries (e.g. crossword or Scrabble), text-to-speech, automatic speech recognition, and optical character recognition/correction.

2.3.2 Structure of nominals: lexc and twolc

With that minimal introduction to `lexc` and `twolc`, I will now present some of the most salient features of the structure of UDAR's lexicon. It should be stressed that the following techniques for modeling Russian are not intended to satisfy any linguistic theory. The decisions underlying the underlying word structure produced by `lexc` and the operations of the orthographic/(morpho-)phonological rules within the two-level formalism are usually informed by formal linguistic descriptions of Russian, but ultimately these decisions are a matter of engineering, and not theory.

One connection to theoretical linguistics that pervades all two-level rules is the relation to source- and product-oriented generalizations (Bybee, 2003). Rules that reference the upper side of two-level characters are essentially making source-oriented generalizations, whereas rules that refer to the lower side of two-level characters are making product-oriented generalizations.

Unfortunately, supplying a complete overview of the structure of Russian is outside the scope of this chapter. I do provide basic explanations and examples of

forms that are being modeled, but only those properties that are relevant to considerations of designing the model are explained. The implications of the model are only discussed when they have relevance to linguistic theory.

Stem shape

A prominent feature of Russian phonology is consonant palatalization (commonly referred to as *hardness* vs. *softness*). Russian orthography marks consonant hardness or softness by two parallel sets of vowels (and other symbols), so that hard consonants are followed by one set, and soft consonants the other. Table 2.2 shows singular forms of two lexemes: *rabota* 'work'(hard *t*) and *milâ* 'mile'(soft *l*).[16] The underlying forms, as they are generated by `lexc`, are given in parentheses.

	'work'	'mile'
NOM	`rabota (rabota>a)`	`milâ (milâ>a)`
ACC	`rabotu (rabota>u)`	`milû (milâ>u)`
GEN	`raboty (rabota>y)`	`mili (milâ>y)`

Table 2.2: Two nouns of the same declension class with different stem palatalization. The underlying `lexc` forms are in parentheses.

The stem boundary, `>`, is added in `lexc` as a frame of reference for `twolc` rules. The stem boundary is always deleted. Additionally, any vowel, soft sign ('), or *j* directly preceding the stem boundary is also deleted. The letter deleted before the stem boundary serves as a reference for whether or how to transform the ending itself. As can be seen in Table 2.2, both *rabota* and *milâ* belong to the same continuation class, i.e. both have identical endings in the `lexc` output. However, because they have different classes of vowels deleted before the stem boundary, their endings are realized differently.

Table 2.3 shows the correspondences defined by the relevant `twolc` rules. This table is read by comparing each character on the so-called upper side (underlying form) with the corresponding character on the so-called lower side (surface form). If a character is deleted, then it appears as either an underscore (_) or zero (0) on the lower side.[17] The first example, `rabota>a`, shows that `twolc` deletes the `a` and `>`.

[16] For two-level rules, I use ISO9 transliteration from Cyrillic to Latin, as it exhibits one-to-one mappings of letters, rather than digraphs which exist in most other transliteration systems. The ISO9 transliteration system is given in the Foreword. The Foreword also contains the Scholarly transliteration system, which I use in contexts other than two-level rules.

[17] In `twolc`, a removed character is represented by a zero, but I sometimes use underscores to improve horizontal readability.

	rabota 'work.SG-NOM'	*rabotu* 'work.SG-ACC'	*raboty* 'work.SG-GEN'
upper	rabota>a	rabota>u	rabota>y
lower	rabot__a	rabot__u	rabot__y

	milâ 'mile.SG-NOM'	*milû* 'mile.SG-ACC'	*mili* 'mile.SG-GEN'
upper	milâ>a	milâ>u	milâ>y
lower	mil__â	mil__û	mil__i

Table 2.3: Upper- and lower-side correspondences for nominal palatalization

All two-level rules operate simultaneously, so the context of each rule must refer to both the upper and lower side of each symbol. The `twolc` formalism treats each pair of upper/lower characters as one unit, separated by a colon, so in the first column of Table 2.3 is represented as
"`r:r a:a b:b o:o t:t a:0 >:0 a:a`". The fourth column is represented as "`m:m i:i l:l â:0 >:0 a:â`". This notation can be abbreviated when both sides are identical, so `m:m` can be simply be written as `m`. Likewise, one side can be underspecified, so `a:a`, `a:â`, and any other declared symbols that include `a` on the upper side can be written as `a:`. There are two relevant `twolc` rules for these examples. The first is responsible for deleting the vowel, soft sign, or *j* before the >. This is formalized in Example (6).[18] This rule changes `a:a` to `a:0` (and `â:â` to `â:0`, etc.) if they are followed by the deleted stem boundary symbol `>:0`.

(6)
```
Vx:0 <=> _ >:0
where Vx in ( a â o ë e j ' )
```

The second `twolc` rule changes the vowel in the ending to match the palatalization of the final consonant of the stem. More formally, this rule changes `a:a` to `a:â` when preceded by the sequence `â:0 >:0`, as shown in Example (7). An expanded version of this rule deals with all the relevant vowels.

(7)
```
a:â <=> â:0 >:0 _
```

[18] For the sake of readability, many of the `twolc` rules given in this dissertation are significantly simplified from the actual rules in UDAR's source files. I also remove unnecessary `twolc` syntax, such as escaping characters, and end of line markers (;).

Spelling rules

As briefly mentioned above, velars (*g*, *k* and *x*), hushers (*ž*, *š*, *ŝ* and *č*), and *c* each have particular constraints on which vowel letters they can be followed by. Zaliznjak's grammatical dictionary defines seven codes to indicate which sets of vowels should be used in inflectional endings. However, since these codes are perfectly aligned with phonological contexts, I discarded Zaliznjak's stem codes, replacing them with `twolc` rules. Examples are given in Table 2.4.

	knigi 'books'	*xorošij* 'good.NOM'	*xorošem* 'good.LOC'	*xožu* 'I walk'
upper	`kniga>y`	`xoroš>yj`	`xoroš>om`	`xož>û`
lower	`knig__i`	`xoroš_ij`	`xoroš_em`	`xož_u`

Table 2.4: Upper- and lower-side correspondences for 'spelling rules'

Fleeting vowels

Many Russian stems have vowels that appear only in specific morphophonological environments, primarily with a zero inflectional ending, or more broadly, when the inflectional ending does not begin with a vowel. Such vowels are specified in the stem, immediately preceded by a special symbol (here, for simplicity, `F`), which is always deleted. Examples are given in Table 2.5.

	okno 'window.NOM'	*okon* 'windows.GEN'	*zemle* 'earth.LOC'	*zemel'* 'earths.GEN'
upper	`okFono>o`	`okFono>`	`zemFelâ>e`	`zemFelâ>'`
lower	`ok__n__o`	`ok_on__`	`zem__l__e`	`zem_el__'`

Table 2.5: Upper- and lower-side correspondences for fleeting vowels

The basic rule for removing fleeting vowels of this type is simple: delete the fleeting vowel if it is followed by an inflectional ending that begins with a vowel on the lower side. This rule is given in Example (8), which says to delete a vowel if it is immediately after `F`, followed by one or more letters (`Letter` is previously defined in the source code), followed by a deleted vowel, *j*, or soft sign, followed by a deleted stem boundary, followed by any symbol with a vowel on the lower side.

(8)
```
Vx:0 <=> F:0 _ Letter+ [V:0|j:0|':0] >:0 :V
where Vx in ( a e ë i o â )
```

This formulation of the rule is robust with less canonical instances of fleeting vowels. For example, *lûbov'û* 'love.INST', does not delete the fleeting vowel because the first symbol after the stem boundary is the soft sign *'*.

There are some important benefits to specifying the vowel quality and stress of all fleeting vowels. Although the vowel quality of most fleeting vowels is predictable—*o*/*ë*/*e*, depending on the context—there are several exceptions, such as *zaâc* 'hare', which has a fleeting *â*. Similarly, the rules given by Zaliznjak for placing stress on fleeting vowels are generally robust, but there are many exceptions, such as *zemlâ~zemél'*, for which the rules predict the form *zémel'*. Since the quality and stress of fleeting vowels is specified for every lexeme, such instances do not pose a problem.

Some additional rules are needed to properly describe instances in which *j* or *'* directly border the fleeting vowel. Take *kopejka~kopeek* 'kopeck', for example. When the fleeting vowel is deleted, the letter *j* is present, but when the fleeting vowel is present, the letter *j* is deleted.[19] Table 2.6 shows the correspondences of these examples.

	kopejka	*kopeek*
	'kopeck.SG-NOM'	'kopeck.PL-GEN'
upper	`kopejFeka>a`	`kopejFeka>`
lower	`kopej__k__a`	`kope__ek__`

Table 2.6: Upper- and lower-side correspondences for fleeting vowels in the lexeme *kopejka* 'kopeck'

A formal description of this relation is given in (9), which says to delete *j* if it immediately precedes `F` followed by a symbol that is a vowel on both the upper and lower sides (i.e. not deleted).

(9) `j:0 <=> _ F:0 V`

A nearly identical situation arises with a number of words in which a soft sign is present when the fleeting vowel is deleted, but the soft sign is removed when the fleeting vowel is present. Table 2.7 gives examples of such words.

The formal description of these correspondences is almost identical to (9), as shown in (10).

(10) `':0 <=> _ F:0 V`

A similar circumstance arises with many nouns with fleeting vowels whose stems end with a phonetic yod, orthographically represented as *j*, *'*, or as part of a com-

[19] The phonetic yod is still implied by the letter *e* following a vowel.

	lëd ‘ice.SG-NOM’	*l'da* ‘ice.SG-GEN’
upper	`l'Fëd`	`l'Fëd>a`
lower	`l__ëd`	`l'__d_a`

Table 2.7: Upper- and lower-side correspondences for fleeting vowels in the lexeme *lëd* ‘ice’

plex vowel, depending on the environment. Russian orthography is such that after vowels, a phonetic yod is represented as *j*, unless it is followed by another vowel, in which case it is represented by the following vowel letter, such as with *kopejka~kopeek* in Table 2.6 above. After consonants, yod is represented by *′*. For lexemes in which this environment alternates—such as those given in Table 2.8 below—the `lexc` lexicon gives the underlying form with *j*, and this is changed to *′* when necessary.

	muravej ‘ant.SG-NOM’	*murav'û* ‘ant.SG-DAT’	*kop'ë* ‘spear.SG-NOM’	*kopij* ‘spear.PL-GEN’
upper	`muravFej`	`muravFej>u`	`kopFijë>o`	`kopFijë>`
lower	`murav_ej`	`murav__'_û`	`kop__'__ë`	`kop_ij__`

Table 2.8: Upper- and lower-side correspondences for fleeting vowels in yod stems, such as *muravej* ‘ant’ and *kop'ë* ‘spear’

A formal description of this relation is given in (11), which says to change *j* to *′* when it is preceded by a consonant, then `F`, then a deleted vowel, and when it is optionally followed by a deleted *e*, *ë*, or *â*, followed by a deleted stem boundary, followed by any symbol with a vowel on the lower side (i.e. not deleted).

(11) `j:' <=> C: F:0 V:0 _ ([e:0|ë:0|â:0]) >:0 :V`

At this point, some of the underlying forms are beginning to look very different from the actual surface representations. This raises the question of how far one should go to capture regularities in the language with two-level rules, as opposed to simply creating new continuation classes in `lexc` to generate forms that are already well-formed. This is mostly a matter of preference, but I have generally used the following guideline. If a form deviates in a way that can be captured by systematic rules, then I encode that deviation in two-level rules, as long as the necessary information could be easily extracted from Zaliznjak’s dictionary. The alternative, which is to capture stem alternations using a much larger set of continuation classes, is certainly viable, and has, in fact, been implemented (e.g.

Vilkki, 1997, 2005).

Adjusting noun inflectional endings

In nouns with stems ending in *-ij*—such as *kafeterij* 'cafeteria', *povtorenie* 'repetition', or *Rossiâ* 'Russia'—there is a regular inflectional deviation from other nouns. In such words, any inflectional ending that is an underlying *e* is realized as an *i*, as shown in Table 2.9.

	kafeterii 'cafeteria.SG-LOC'	*povtorenii* 'repetition.SG-LOC'	*Rossii* 'Russia.SG-LOC/DAT'
upper	`kafeterij>e`	`povtorenie>e`	`Rossiâ>e`
lower	`kafeteri__i`	`povtoreni__i`	`Rossi__i`

Table 2.9: Upper- and lower-side correspondences for *e*-inflection in i-stems: *kafeterij* 'cafeteria', *povtorenie* 'repetition', and *Rossiâ* 'Russia'

This is formally expressed using the `twolc` statement given in (12), which states that *e* changes to *i* when it is preceded by *i*, followed by a deleted *j*, *ë*, *e*, or *â*, followed by a deleted stem boundary.

(12) `e:i <=> i [j:0|ë:0|e:0|â:0] >:0 _`

Word stress

One of the primary motivations for developing UDAR was to have a working transducer with stressed wordforms. This section will demonstrate how this is achieved. Discussion of previous rules deliberately ignored word stress for the sake of simplicity. However, from this point on, it should be remembered that the UDAR's two-level rules treat stressed and unstressed vowels as separate entities. For example, a rule that refers to *a* cannot be assumed to also refer to *á*, neither can *â* be assumed to refer to *ấ*, etc.

Russian has several complex patterns of shifting stress, which would pose a serious complication for a system limited to continuation classes. Table 2.10 illustrates one of these stress patterns.

Representing such patterns of shifting stress in the `lexc` language would require a continuation class specific to the final string of letters of each lexeme (e.g. *-uka*). Example (13) shows the `lexc` code that would be needed for the nominative and accusative singular forms of *ruka*.

(13) a. `ruka:r UKA_f' ;`

	SG	PL
NOM	*ruká*	*rúki*
ACC	*rúku*	*rúki*
GEN	*rukí*	*rúk*
LOC	*ruké*	*rukáx*
DAT	*ruké*	*rukám*
INS	*rukój*	*rukámi*

Table 2.10: Shifting stress pattern of *ruka* 'hand'

b.
```
LEXICON UKA_f'
+Nom:uká # ;
+Acc:úku # ;
```

The entry in (13-a) can only specify the `r` of the stem, since the following letter varies between `u` and `ú`. The continuation class in (13-b) then completes the stem and also adds the ending. The necessity to create continuation classes specific to the final string of letters would likely require thousands more continuation classes than an FST without stress. However, because of the particulars of Russian's system of stress, `twolc` provides the means for an elegant solution. With only a small handful of exceptions, a Russian lexeme can have a maximum of three positions of stress in any of its forms: a stem vowel, a fleeting vowel at the end of the stem, and a vowel in the ending.

The `lexc` code of UDAR marks the stress on all vowels in the stem that are stressed in one form or another. Each inflection class then has multiple variants of `lexc` continuation classes for possible stress patterns. What this means is that the presence or absence of stress in the ending of an underlying form is definitive, whereas stress marked on the stem should only be realized if the ending is not stressed. This alternative to (13) would result in the following wordforms (simplified for demonstration): `rúk>á` and `rúk>u`. The `twolc` rule then removes all but the rightmost stress: `ruká` and `rúku`. A full-fledged example of this is shown in Table 2.11, using *sestrá* 'sister.SG-NOM', *sestrý* 'sister.SG-GEN' and *sëstry* 'sister.PL-NOM/ACC'. Note that stress is usually marked with an acute accent (´), but in the case of the letter *ë* it is marked with an umlaut/diaeresis diacritic (¨).[20]

In the example of *sestra*, the stress is removed from the first *ë* because the final *a* is stressed. The fleeting vowel is deleted according to the fleeting vowel rule discussed previously. In the example of *sëstry*, the ending is unstressed and the

[20]The orthography and phonology of *ë* are closely connected. When *ë* is unstressed, it is written and pronounced exactly like *e*. In Contemporary Standard Russian, *ë* is typically written without the diacritic, even when it is stressed.

	sestra 'sister.SG-NOM'	*sëstry* 'sister.PL-NOM'	*sestër* 'sister.PL-ACC/GEN'
upper	`sëstFërá>á`	`sëstFërá>y`	`sëstFërá>`
lower	`sest__r__á`	`sëst__r__y`	`sest_ër__`

Table 2.11: Upper- and lower-side correspondences for word stress, example word *sestra* 'sister'

fleeting *ë* is deleted, so the stress on the first *ë* remains. In the example of *sestër*, there is a zero ending (which by definition cannot take stress), which means that the fleeting vowel is realized in the surface form. And since the fleeting vowel is stressed, the stress is removed from the first *ë*. The examples given in Table 2.11 show how two-level rules can be used to generate forms with correct stress placement, all based on the same underlying stem.

There are a handful of lexemes that cannot be represented within this system, most notably the multisyllable-stemmed feminine nouns with Zaliznjak's code `f′`, such as *golova* 'head' and *skovoroda* 'frying pan'. When the stress falls on the stem of such words, it is on the first syllable, except the genitive plural, which is stressed on the last syllable of the stem. Since they can have stress on more than one syllable in the stem, they violate one of the enabling assumptions of the two-level rule described above. Likewise, a very small number of other nouns violate this assumption in other ways. The lexeme *ozero* 'lake' is stressed on the the first syllable throughout the singular, but the second syllable through the plural. This alternation of stress position *within* the stem is exceptional. All such forms are generated by using additional continuation classes in the `lexc` code.

Genitive plural of nouns

The inflectional ending of the genitive plural of nouns is morphophonologically conditioned, and can be one of three possibilities: *-ov*/*-ëv*/*-ev*, zero/*-'/j*, or *-ej*. Each of these possibilities will be discussed in more detail below. All declension classes follow the same rules for determining which inflection appears, so rather than multiply continuation classes, I captured this regularity using two-level rules. The `lexc` code assigns only one genitive plural ending: *-ov* (or *óv*, if stressed). When necessary, this ending is transformed into the appropriate allomorph, as demonstrated in the following tables. The general approach echoes that of Halle (1994), who claims that all declension classes have the same genitive plural inflection: the back yer (-ъ). Halle's work claims that the allomorphy in genitive plural inflections is actually a complex form of phonologically determined allomorphy.

Lexemes with exceptional genitive plural endings—of which there are many—

are simply hard-coded in the `lexc` code. Regarding nouns with gaps in the genitive plural, such as *mečt* '?dreams.GEN-PL', I simply follow the codes given in Zaliznjak's dictionary. In this particular case, *mečt* is given the `+Prb` tag, which indicates that it is problematic.

The genitive plural endings *-ov*/*-ëv*/*-ev* appear on nouns whose nominative singular ends in a consonant.[21] Table 2.12 shows nouns with the inflection *-ov*. Any changes to the vowel in the ending of these nouns are due to palatalization and spelling rules, the rules for which have already been discussed in Table 2.3 and Table 2.4 above.

	polóv 'floor.PL-GEN'	*murav'ëv* 'ant.PL-GEN'	*tórtov* 'cake.PL-GEN'	*zájcev* 'hare.PL-GEN'
upper	`pól>óv`	`muravFéj>óv`	`tórt>ov`	`zájFâc>ov`
lower	`pol_óv`	`murav__'_ëv`	`tórt_ov`	`záj__c_ev`

Table 2.12: Upper- and lower-side correspondences for genitive plural inflection *-ov*/*-ëv*/*-ev*

Table 2.13 shows examples of nouns whose genitive plural inflectional ending is zero. Most nouns with a vowel in the nominative singular ending belong to this category.

	dél	*zdánij*	*nedél'*	*podúšek*
upper	`délo>óv`	`zdánie>ov`	`nedFélâ>ov`	`podúšFeka>ov`
lower	`dél_____`	`zdáni__j_`	`ned_él__'_`	`podúš_ek_____`

Table 2.13: Upper- and lower-side correspondences for genitive plural zero ending

In all such lexemes, the final *v* in the *-ov* sequence on the upper side is deleted. However, the *o* can change in three different ways. If the stem ends in a hard consonant, then the *o* is simply deleted, as in the case of *del* and *podušek*. If the stem ends in a vowel, then the *o* is changed to a *j*, as in *zdanij*. If the stem ends in a soft consonant, then the *o* is changed to a *′*, which marks softness in the absence of a vowel. These relations are formalized in the following rules. The rules for simple deletion are given in (14), which is divided into two parts for legibility. The expression in (14-a) states that *o* and *ó* are deleted when they are preceded by a velar (*g*/*k*/*x*) or a so-called paired consonant (those consonants with phonemic hard and soft variants), followed by a deleted *a*, *á*, *o* or *ó*, followed by a deleted stem

[21] For nouns lacking a nominative singular—such as pluralia tantum—a hypothesized stem is specified in `lexc` so that the appropriate genitive plural is produced. For example, *naušniki* 'headphones' is assigned the stem `naušnik`, which yields the genitive plural *naušnikov*.

boundary, and when they are followed by a deleted *v*. The expression in (14-b) states that *o* and *ó* are deleted when they are preceded by a husher (*ž*/*š*/*ŝ*/*č*) or *c*, followed by a deleted *a*, *á*, *e* or *ó*, followed by a deleted stem boundary, and when they are followed by a deleted *v*. The only difference between (14-a) and (14-b) is that the deleted vowels before the stem boundary include *o* and *e*, respectively. This is a trivial fact that is the result of spelling rules, as discussed above in Table 2.4: an underlying unstressed *o* after hushers and *c* is spelled *e*.

(14) `[o:0|ó:0] <=>`

a. `[g|k|x|b|v|d|z|l|m|n|p|r|s|t|f]`
`[a:0|á:0|o:0|ó:0] >:0 _ v:0`

b. `[ž|š|ŝ|č|c] [a:0|á:0|e:0|ó:0] >:0 _ v:0`

The rule for changing *o* to *j* is given in (15), which states that *o* and *ó* change to *j* when they are preceded by a vowel, followed by a deleted *â*, *ấ*, *e*, *é* or *ë*, and when followed by a deleted *v*.

(15) `[o:j|ó:j] <=> V: [â:0|ấ:0|e:0|é:0|ë:0] _ v:0`

The rule for changing *o* to *'* is given in (16), which states that *o* and *ó* change to *'* when they are preceded by a paired consonant, followed by a deleted *a*, *á*, *e* or *ó*, followed by a deleted stem boundary, and when they are followed by a deleted *v*.

(16) `[o:'|ó:'] <=> [b|v|d|z|l|m|n|p|r|s|t|f]`
`[â:0|ấ:0|e:0|é:0|ë:0] _ v:0`

In addition to the more canonical rules given in (14), (15) and (16), there are a few exceptional contexts in which *o* is deleted. Examples of these exceptional contexts are given in Table 2.14.

	bášen 'tower.PL-GEN'	*seméj* 'family.PL-GEN'	*sekvoj* 'sequoia.PL-GEN'
upper	`báš Fenâ>ov`	`sém Féjấ>ov`	`sekvójâ>ov`
lower	`báš_en____`	`sem_éj____`	`sekvój____`

Table 2.14: Upper- and lower-side correspondences for genitive plural zero ending for *bašnâ* 'tower', *sem'â* 'family', and *sekvojâ* 'sequoia'

In the first example, *bášen*, we would expect the *o* to be changed to *'*, according to the rule in (16). However, in the context expressed in (17-a), the stem-final *n* becomes hard in the genitive plural. This rule states that *o* and *ó* are deleted when they are preceded by a deleted `F`, followed by a vowel that is not deleted, followed by an *n*, followed by a deleted *â*, followed by a deleted stem boundary, and when

they are followed by a deleted *v*.[22] It should be noted that this context overlaps with the rule given in (16), which requires that rule to be rewritten to avoid a conflict between the two. For the sake of simplicity, the revised rule is not given here, but it can be found in the `twolc` source code.[23]

The final two rules in (17) are more straightforward. In fact, they are not exceptional in any way, other than that they are simply not canonical cases of zero genitive plural endings. The behavior of *sem'â* 'family' (rule given in (17-b)) follows directly from the fleeting vowel rules given above in (8) and (11). The behavior of *sekvojâ* 'sequoia' (rule given in (17-c)) is only unusual because it represents a very small number of foreign borrowings that maintain the sequence `V j V` in many of its surface forms. However, in the genitive plural, it behaves as expected, simply dropping the final vowel.

(17) `[o:0|ó:0] <=>`

a. `F:0 V: n â:0 >:0 _ v:0`
b. `F:0 V: j: [â:0|ấ:0|e:0|ë:0] >:0 _ v:0`
c. `\F:0 V ( F:0 V: ) j [â:0|ấ:0|e:0|ë:0] >:0 _ v:0`

Finally, we turn to the *-ej* inflection for genitive plural nouns. This inflectional ending is found in lexemes whose stem ends in either a husher (*ž*/*š*/*ŝ*/*č*) or soft paired consonant, marked by a soft sign (*'*) or one of the following soft-indicating vowels: *â*, *e*, or *ë*. There are two special cases of this rule. If the nominative singular of the lexeme ends in a consonant, then *-ej* is used regardless of stress position. On the other hand, if the nominative singular of the lexeme ends in a vowel, then *-ej* is used only if the gentive plural ending is stressed. Examples of such words are given in 2.15.

Part of the change from *-ov* to *-ej* is already covered by the palatalization and spelling rules covered above, which changes an unstressed *o* to *e* after hushers and soft consonants. This change can be seen in the example of *mátčej*. However, a new rule is needed to change *ó* to *é*, as shown in (18). The first rule states that *ó* changes to *é* when it is preceded by a husher, followed by a deleted *a*, *á*, *e*, *ó*, or *'*, followed by a deleted stem boundary, and when followed by a *v* that changes to a *j*. The second rules states that *ó* changes to *é* when preceded by a paired consonant, followed by a deleted *e*, *é*, *ë*, *â*, *ấ*, or *'*, followed by a deleted stem boundary, and when followed by a *v* that changes to a *j*.

[22] In Zaliznjak's dictionary, this context is signified by the code ж `2*a`.

[23] Currently at `https://victorio.uit.no/langtech/trunk/langs/rus/src/phonology/rus-phon.twolc`

	konéj ‘horse.PL-GEN’	*mátčej* ‘match.PL-GEN’
upper	kón'>óv	mátč>ov
lower	kon__éj	mátč_ej

	levšéj ‘left-hander.PL-GEN’	*moréj* ‘sea.PL-GEN’
upper	levšá́>óv	móre>óv
lower	levš__éj	mor__éj

Table 2.15: Upper- and lower-side correspondences for genitive plural *-ej* ending for *kon'* ‘horse’, *matč* ‘match’, *levša* ‘left-hander’, and *more* ‘sea’

(18) ó:é <=>

a. [ž|š|ŝ|č] ([a:0|á:0|e:0|ó:0|':0]) >:0 _ v:j

b. [b|v|d|z|l|m|n|p|r|s|t|f]
[e:0|é:0|ë:0|â:0|á́:0|':0] >:0 _ v:j

Since the two rules in (18) are both dependent on the presence of the v:j pair, these rules are only applied in the context in which the rules given in (19), which express the contexts in which *v* changes to *j*. The contexts are identical, but divided differently. Rule (19-a)—which covers the form *matčej* from Table 2.15—states that *v* changes to *j* when preceded by a husher, optionally followed by a deleted soft sign, followed by a deleted stem boundary, followed by an *o* or *ó* on the upper side. Rule (19-b)—which covers the form *konej* from Table 2.15—states that *v* changes to *j* when preceded by a paired consonant, followed by a deleted soft sign, followed by a deleted stem boundary, followed by an *o* or *ó* on the upper side. The rule given in (19-c)—which covers the wordform *levša* from Table 2.15—states that *v* changes to *j* when preceded by a husher, followed by a deleted *a*, *á*, *e* or *ó*, followed by a deleted stem boundary, followed by an *ó* on the upper side. And lastly, rule (19-d)—which covers the wordform *more* from Table 2.15—states that *v* changes to *j* when preceded by a paired consonant, followed by a deleted *e*, *é*, *ë*, *â* or *á́*, followed by a deleted stem boundary, followed by an *ó* on the upper side.

(19) v:j <=>

a. [ž|š|ŝ|č] (':0) >:0 [o:|ó:] _

b. [b|v|d|z|l|m|n|p|r|s|t|f] ':0 >:0 [o:|ó:] _

c. [ž|š|ŝ|č] [a:0|á:0|e:0|ó:0] >:0 ó: _

d. [b|v|d|z|l|m|n|p|r|s|t|f]
[e:0|é:0|ë:0|â:0|á́:0] >:0 ó: _

Comparative adjectives

Short form comparative adjectives have some features that are effectively encoded as two-level rules. The comparative is distinguished by the inflectional ending *-ee* (or *ée*, if stressed). For example, *nóvyj* 'new' has the comparative *novée* '(is) newer'. However, adjectives with velar stems undergo the following changes in the comparative. The final velar—*g*, *k* or *x*—mutates to *ž*, *č* or *š*, respectively, and the inflection ending is truncated to an unstressed *e*. If the underlying (upper-side) ending is stressed, then the stress shifts left to the last syllable of the stem. Examples of the comparative are given in Table 2.16.

	staree 'old.COMP'	*stróže* 'strict.COMP'	?*odinóče* 'single.COMP'	*tíše* 'quiet.COMP'
upper	`stár>ée`	`stróg>ée`	`odinók>ee`	`tíx>ée`
lower	`star_ée`	`stróž_e_`	`odinóč_e_`	`tíš_e_`

Table 2.16: Upper- and lower-side correspondences for comparative adjectives

Several rules are needed to render these wordforms correctly. The rule given in (20) defines the mutation from velar to husher. It states that *g* changes to *ž*, *k* to *č*, and *x* to *š*, when they are followed by a deleted stem boundary, followed by a *e* or *é* on the upper side, followed by a *e* on the upper side.

(20) `[g:ž|k:č|x:š] <=> _ >:0 [e:|é:] e:`

The deletion of the final *e*[24] is expressed in rule (21), which states that *e* is deleted when it is preceded by a velar on the upper side, followed by a deleted stem boundary, followed by a *e* or *é* on the upper side.

(21) `e:0 <=> [g:|k:|x:] >:0 [e:|é:] _`

Finally, in adjectives whose comparatives are assigned a stressed ending, such as *stróže* and *tíše*, it is necessary to destress the first *e* in the inflectional ending. This is defined by the rule given in (22), which states that *é* becomes *e* when it is preceded by a velar on the upper side, followed by a deleted stem boundary, and when it is followed by a *e* on the upper side.

(22) `é:e <=> [g:|k:|x:] >:0 _ e:`

In addition to the forms given in Table 2.16, adjectives also have a so-called atten-

[24] In fact, the `lexc` outputs two comparative endings: *-ee* and *-ej*. The latter is used in rapid speech or poetry, and it is ignored in the present discussion for the sake of simplicity. The rule to remove the *j* is virtually identical to the rule in (21).

uated comparative,[25] which is not included in Zaliznjak's description of comparatives (Zaliznjak, 1977, p. 58). The attenuated comparative is identical to the standard comparative, but with the prefix *po-* added, e.g. *postarée*, *postróže*, or *potíše*. Attenuated comparatives can be used adverbially (23-a), predicatively (23-b), or attributively (23-c).[26]

(23) a. *V Rossii igraût čut' pobystree, čem v Grecii.*
In Russia play.3PL slightly fast.CMPR-ATTEN than in Greece
'In Russia, they play just a little faster than in Greece.'

b. *Èto poxuže tvoix problem.*
This worse.ATTEN your.GEN problems.GEN
'This is a little worse than your problems.'

c. *Zdes' rešali lûdi poumnee tebâ.*
Here resolve.PAST people.NOM smart.CMPR-ATTEN you.GEN
'This was resolved by people a little smarter than you.'

The formation of such comparatives is problematic for the left-to-right nature of lexicon construction in `lexc`, because the prefix is only present in this one cell of the paradigm. However, this is easily resolved with the use of a two-level rule. I declare two archiphonemes—`P` and `A`— that are prepended to the entire adjective lexicon. By default, `P` and `A` are always deleted, so they have been ignored in the descriptions of all other nominal rules in this chapter for the sake of simplicity. However, the rules governing these archiphonemes are given here. `P` and `A` are realized as *p* and *o*, respectively, whenever the wordform ends with a `P`, as well. Examples are given in Table 2.17.

	novée 'newer'	*ponovée* 'a little newer'
upper	`PAnóv>ée`	`PAnóv>éeP`
lower	`__nov_ée`	`ponov_ée_`

Table 2.17: Upper- and lower-side correspondences for comparatives with prefix *po-*

The rules needed to realize `P` and `A` as surface letters are given in (24). The first rule states that `P` changes to *p* when it is at the beginning of the word and followed by `A` on the upper side, followed by any sequence of any symbols, followed by an upper-side *e* or *j*, followed by a deleted `P` at the end of the word.

[25] Sometimes these forms are also referred to as "prefixed" comparatives.

[26] A more thorough investigation of the semantics and syntax of attenuated comparatives is available in Boguslavsky and Iomdin (2009), or at `http://rusgram.ru/Сравнительная_степень_на_по-`.

(24) a. `P:p <=> .#. _ A: ?* [e:|j:] P:0 .#.`
b. `A:o <=> .#. P: _ ?* [e:|j:] P:0 .#.`

Masculine short-form adjectives

Most Russian adjectives have four so-called short-forms: masculine, neuter, feminine and plural. These four forms are primarily used predicatively. The inflectional endings of these four forms are basically the same an the canonical noun endings for each singular gender and plural: zero for masculine, *-o* for neuter, *-a* for feminine, and *-y* for plural. The actual surface realizations of these endings can change according to palatalization and spelling rules discussed previously, and for the most part, these changes are already captured by rules given above. However, in order to avoid complicating existing rules unnecessarily, I created a new symbol `Z` to render the zero ending of the masculine short-form. Examples of the masculine short-form are given in Table 2.18.

	nóv 'new'	*izlíšen* 'excessive'	*sín'* 'blue'	*dlinnošéj* 'long-necked'
upper	`nóv>Z`	`izlíšFen'>Z`	`sín'>Z`	`dlinnošéj>Z`
lower	`nóv__`	`izlíš_en___`	`sín__'`	`dlinnošé__j`

Table 2.18: Upper- and lower-side correspondences for masculine short-form adjectives

The first example, *nov* is very straightforward, and its pattern is described in rule (25-a). This rule states that `Z` is deleted when it is preceded by a velar, husher, paired consonant or *j*, followed immediately by a deleted stem boundary. The second example, *izlišen*, is very similar to the example of *bašnâ* discussed in Table 2.14 and rule (17-a), above. Its changes are expressed in rule (25-b), which states that `Z` is deleted when it is preceded by a deleted `F`, followed by a vowel, followed by an *n*, followed by a deleted soft sign, followed by a deleted stem boundary. This is exceptional because in such lexemes, the *n* is soft in every form except the masculine short-form.

(25) `Z:0 <=>`
a. `[g|k|x|ž|š|ŝ|č|c|b|v|d|z|l|m|n|p|r|s|t|f|j] >:0 _`
b. `F:0 V: n ':0 >:0 _`

The third example, *sin'*, is what we typically expect of a soft stem, and its changes are expressed in the rules in (26), which collectively define all soft paired consonant adjective stems other than the one captured in (25-b). Rule (26-a) captures every

paired consonant other than *n*. Rules (26-b) and (26-c) capture contexts with *n* that differ from (25-b). More specifically, (26-b) captures contexts in which the preceding vowel is *not* a fleeting vowel—such as *sin'* 'blue' above—and (26-c) defines contexts when *n* is preceded by a consonant. Currently, only one lexeme in the lexicon is captured by this rule: *verxnij~verxn'* 'upper'.

(26) `Z:'<=>`

a. `[b|v|d|z|l|m|p|r|s|t|f] ':0 >:0 _`
b. `\F:0 V n ':0 >:0 _`
c. `C n ':0 >:0 _`

The last example in Table 2.18—*dlinnošej* 'long-necked'— represents a very small set of adjectives whose stems do not contain a fleeting vowel, and the last letter is *j*. Currently, only seven lexemes are covered by this rule, all of them based on the root *ŠEJ* 'NECK'. Rule (27) states that `Z` changes to *j* when it is preceded by any character other than `F`, followed by a vowel, followed by a *j* on the upper side, followed by a deleted stem boundary.

(27) `Z:j <=> \F:0 V j:  >:0 _`

The preceding discussion has introduced the two-level formalism as implemented in the `lexc` and `twolc` languages. I have also given an overview of the most relevant two-level rules governing the structure of Russian nominals, including stem palatalization and stress placement. In the following section, I discuss some of the two-level rules specific to Russian verbs.

2.3.3 Structure of verbs: lexc and twolc

There are two prevailing approaches to formal descriptions of Russian verb conjugation: the one-stem system and the two-stem system.[27] The two-stem approach asserts that each verb has two separate surface stems between which no regular relation is assumed. One stem is used to form the infinitive and past, and the other is used to form present, future and imperative forms. The one-stem system, on the other hand, posits an underlying stem from which the two separate stems can be derived. UDAR is primarily based on Zaliznjak's *Grammatical dictionary of Russian*, which favors a two-stem system, so UDAR's verbal lexicon is also structured according to a two-stem system.

Note that for the sake of simplicity in the two-level rules, a different symbol is used to mark the stem boundary for verbal morphology (<), as opposed to the nominal stem boundary symbol (>). The stem boundary symbols merely serve

[27] For a discussion of these two systems, see Chvany (1990) or Nesset (2008, ch. 5).

as a trigger for the application of a certain set of rules, so using a separate symbol for verbs removes the need to write the rules to accommodate one another (i.e. avoiding overlapping contexts). In reality, these rules are not strictly limited to either nominals or verbs. There are instances where the 'nominal' stem boundary `>` is used within the verbal system to trigger the application of some rules discussed above. For example, the verb *tancevat'* 'dance' has the upper-side form `tanc>ová<t'`, which takes advantage of the spelling rules defined for nominals in order to unify its continuation class with verbs like *čuvstvovat'* 'feel'. In terms of linguistics, one could say that the so-called nominal rules also apply in word formation.

Stem alternations

Russian exhibits stem-final consonant alternations in certain forms of the nonpast conjugations and related verb forms. There are two different types of stem alternations in Russian verbs: stems ending in a labial gain an epenthetic *l*, and certain other consonant letters change to a husher, as shown in (28). Note that two consonants' alternation patterns are nondeterministic: *d* alternates with both *ž* and *žd*,[28] and *t* alternates with both *ŝ* and *č*. In order to render these alternations, I declare two archiphonemes `D` and `T` to stand in for *d* and *t* in stems with the respective *žd* and *ŝ* alternations.

(28)
g	d	D	z	s	x	sk	st	T	t	k	b	v	m	p	f
ž	ž	žd	ž	š	š	ŝ	ŝ	ŝ	č	č	bl	vl	ml	pl	fl

The `lexc` code marks where alternations should occur by means of a special symbol (abbreviated here as `M`, for simplicity).[29] In the case of labial stems, the `M` is converted to an *l*. With the archiphoneme `D`, the `M` changes to a *d*. For the remaining consonants listed in (28), the `M` is deleted and the preceding consonant is converted to its corresponding husher. Examples of this are given in Table 2.19.

	lûblű '(I) love'	*pláčet* 'cries'	*iŝú* '(I) seek'
upper	`lûbM<ű`	`plákM<et`	`ískM<ú`
lower	`lûbl_ű`	`pláč__et`	`i_ŝ__ú`

Table 2.19: Upper- and lower-side correspondences for verbal stem mutations

[28] The *d~žd* alternation only occurs in the past passive participle.

[29] `M` stands for 'mutation'.

The first example in Table 2.19, *lûblû*, is an example of a labial stem that gains an epenthetic *l*. This is described in rule (29), which states that `M` changes to *l* after labial consonants.

(29) `M:l <=> [b|p|v|f|m] _`

Further, the second example, *pláčet*, is one in which the stem-final consonant alternates with a husher. This type of alternation is captured by the rules in (30)-(32). Rule (30) simply states that each of the listed alternations occurs when followed by the symbol `M` on the upper side.

(30) `[g:ž|d:ž|D:ž|z:ž|s:š|x:š|T:ŝ] <=> _ M:`

However, the rules for *t* and *k* are slightly different because there are two possibilities, based on their context. The clusters *st* and *sk* alternate with *ŝ*, while every other context of *t* and *k* alternate with *č*. The alternation in verbs like *plačet* from Table 2.19 is covered by rule (31), which states that *t* and *k* change to *č* when they are preceded by anything other than an *s* on the upper side, and when they are followed by a deleted `M`.

(31) `[t:č|k:č] <=> \s: _ M:0`

The complementary context is expressed in rule (32), which captures the stem alternation of *iŝu* from Table 2.19. It simply states that *t* and *k* change to *ŝ* when preceded by an upper-side *s* and followed by a deleted `M`.

(32) `[t:ŝ|k:ŝ] <=> s: _ M:0`

One other stem alternation makes use of the symbol `M`, but without deleting it. As mentioned earlier, a small number of verbs with a stem in *d* alternate with *žd* in the past passive participle. This alternation is achieved by changing the `M` to a *d*, as shown in Table 2.20.

	ubežděnnyj ‘convinced’
upper	`ubeDMënn>yj`
lower	`ubeždënn_yj`

Table 2.20: Upper- and lower-side correspondences for past passive participle stem alternations to *-žd-*

In this alternation, there are two changes. The `D` is changed to *ž* by rule (30), above. It should be noted that the `M` symbol in that rule is not given its maximal

specification `M:0` precisely because of the situation presented here, in which the `M` is not deleted. In participles such as *ubeždënnyj*, the `M` is changed to *d*, as declared in rule (33), which simply states that `M` changes to *d* when it is preceded by a `D` and followed by either *e* or *ë* on the upper side, followed by an *n*, optionally followed by another *n*, followed by a deleted nominal stem boundary. This sequence is a unique signature of the *-i-* conjugation's past passive participle, both long- and short-form.

(33) `M:d <=> D _ [e:|ë:]  n (n) >:0`

One other kind of stem alternation should be mentioned here. A closed class of Russian verbs have stems ending in *g* and *k*, such as *moč'* 'be able / can' and *peč'* 'bake'. Their non-past inflections are shown in example (34).

(34) a. 'can.SG-1' *mogú*
'can.SG-2' *móžeš'*
'can.SG-3' *móžet*
'can.PL-1' *móžem*
'can.PL-2' *móžete*
'can.PL-3' *mógut*
b. 'bake.SG-1' *pekú*
'bake.SG-2' *pečëš'*
'bake.SG-3' *pečët*
'bake.PL-1' *pečëm*
'bake.PL-2' *pečëte*
'bake.PL-3' *pekút*

In such verbs, the letters *g* and *k* change to *ž* and *č* when they are followed by *e* or *ë*. This alternation can be encoded easily without the help of a special symbol, which allows these verbs to use the same `lexc` continuation class as other verbs belonging to the *-e-* conjugation. Two examples of this alternation are given in Table 2.21.

	móžem '(we) can'	*pečëm* '(we) bake'
upper	`móg<em`	`pëk<ëm`
lower	`móž_em`	`peč_ëm`

Table 2.21: Upper- and lower-side correspondences for verbal stem mutations of *moč'* and *peč'*

The rule for this alternation are given in (35), which states that *g* changes to *ž* and *k* to *č*, when they are preceded directly by anything other than a *t*, and followed by a deleted verbal stem boundary, followed by either a *e* or *ë* on the upper side.

(35) `[g:ž|k:č] <=> \t _ <:0 [e:|ë:]`

The reason that there is a constraint to not have a *t* immediately preceding is because of verbs based on the stem *tkat'* 'weave', which do not exhibit this stem alternation, e.g. *tkëm* '(we) weave'. Too broad a constraint, such as disallowing all consonants, would be too limiting, since there are examples like *lgat'* 'lie', which do exhibit this alternation, e.g. *lžëm* '(we) lie'. Unfortunately, there are not enough verbs of this stem type to clearly establish phonetic categories of consonants that do allow the alternation, so the rule merely rules out the only known exception.[30]

Vowels in nonpast endings

There are a number of vowel letter adjustments in the verbal non-past inflectional endings that are easily and efficiently realized using two-level rules. Among the motivations for these rules are both actual morphophonological constraints, as well as normative orthographic rules (which are the remnant of historical phonological constraints, and which actually contradict modern phonology).[31] Ultimately, these two-level rules are designed to model orthographically well-formed words, so the descriptions of the following rules should be understood as descriptions of the orthography, and not necessarily of morphophonology.

The first rule is concerned with the realization of *û* and *â* in verbal endings. This rule applies to the first person singular and the third person plural forms of verbs belonging to the *-i-* conjugation. It is that *û* and *â* change to *u* and *a*, respectively, when they follow a husher. Examples of this are given in Table 2.22.

	slýšu '(I) hear'	*slýšat* '(they) hear'	*xožú* '(I) walk'
upper	`slýš<û`	`slýš<ât`	`xódM<́û`
lower	`slýš_u`	`slýš_at`	`xož__ú`

Table 2.22: Upper- and lower-side correspondences for *û* and *â* in verbal endings

[30] This exception could also be modeled by using special continuation classes for *tkat'* in `lexc`. However, this rule is both simpler to implement, and it is at least potentially a valid description from a theoretical point of view.

[31] Historically, the hushers (*ž*, *š*, *ŝ*, and *č*) and *c* were all palatalized, but *ž*, *š* and *c* have since become non-palatalized. Despite this phonetic difference, and its phonological consequences, these consonant letters are treated the same orthographically.

All three forms given in Table 2.22 are covered by the same rule. Because of the stem alternations discussed in the previous section, the rule must declare a *lower-side* husher, i.e., it is a product-oriented generalization. The rule, given in (36), states that *û*, *ű́*, *â*, and *á́* change to *u*, *ú*, *a*, and *á*, respectively, when they are preceded by a husher on the lower side, optionally followed by a deleted symbol `M`, followed by a deleted verbal stem boundary.

(36) `[û:u|ű́:ú|â:a|á́:á] <=> [:ž|:š|:ŝ|:č] (M:0) <:0 _`

The second rule also applies to the spelling of the first person singular and third person plural inflectional endings, but with verbs in the *-e-* conjugation. As shown above, the `lexc` code assigns the *-û* ending to the first person singular of *-i-* conjugation verbs. However, `lexc` assigns the *-u* ending to verbs of the *-e-* conjugation. This creates an underlying phonological distinction between the two classes of verbs, allowing two-level rules to target only one class at a time. Not only is this convenient from an engineering perspective, it is a reflection of the historical factors that led to the differences in stem palatalization between the verb classes.

In *-e-* conjugation stems ending with a paired consonant, the first-person singular and third-person plural forms have a hard stem (i.e. the ending begins with *-u*, and the remaining forms in the paradigm are soft (their endings all begin with *ë* or *-e*). However, there are three contexts in which the first-person singular and third-person plural forms have endings beginning in *-û*. Examples of these contexts are shown in Table 2.23. The first context is when the stem ends in a phonetic yod. Most commonly, this is when the stem ends in an orthographic vowel—such as in *laû* '(I) bark'—but it also occurs in five verbal roots containing a fleeting vowel: *bit'* 'hit', *vit'* 'wind', *lit'* 'pour', *pit'* 'drink', and *šit'* 'sew'. The second context is when the stem ends in an *l*, as shown in *pošlû* '(I) will send'. The last context is verbs based on two roots— *borot'* and *porot'*—such as *borot'sâ* 'wrestle/fight'. This last category is realized by the use of a special symbol (here `R`, for simplicity), which always changes to *r* in the lower side.

	láû '(I) bark'	*p'űt* '(they) drink'	*pošlű́* '(I) will send'	*borű́s'* '(I) wrestle'
upper	`lá<u`	`pFéj<út`	`póšl<ú`	`bóR<ús'`
lower	`lá_û`	`p__'_űt`	`pošl_ű́`	`bor_ű́s'`

Table 2.23: Upper- and lower-side correspondences for *u* in verbal endings

The rules for these contexts are given in (37).

(37) `[u:û|ú:ű́] <=>`

a. `V <:0 _`
b. `F:0 V: j: <:0 _`
c. `:l <:0 _`
d. `R:r <:0 _`

Rule (37-a) covers the example *laû* from Table 2.23, and states that these changes occur when preceded by a vowel followed by a deleted verbal stem boundary.

Rule (37-b) covers the example *p'ût*, and it states the these changes occur when the vowels are preceded by a deleted `F`, followed by an upper-side vowel, followed by an upper-side *j*, followed by a deleted verbal stem boundary. The yod after a fleeting vowel exhibits the same correspondences as those in Table 2.8 above. This type of verbal stem is also discussed in the section on imperatives below.

Rule (37-c) covers the example *pošlû* from Table 2.23, and it states that these changes occur when preceded by a lower-side *l*, followed by a deleted verbal stem boundary. It is important that the *l* is specified as lower-side, so that it will capture labial stem alternations, such as *dremlû* '(I) snooze' and *koleblûtsâ* '(they) fluctuate'.

Lastly, rule (37-d) covers the example *borûs'* from Table 2.23, and it states that these changes occur when preceded by an `R` that changes to an *r*, followed by a deleted verbal stem boundary.

Imperatives

Russian imperatives pose a similar problem to that of the genitive plural discussed above. The realization of imperative inflectional endings is conditioned by stem shape and stress position, which cuts across the categories created by the current continuation classes in the `lexc` source code. Rather than multiply the number of continuation classes, I handle these inflections with `twolc` rules. This approach simplifies future `lexc` code maintenance by reducing the likelihood of manual miscategorization of new lexemes.

There are several different contexts that determine the realization of the imperative ending. One of the most import factors is stress placement, which the `lexc` code distinguishes by using two separate symbols, `U` for the unstressed ending and `S` for the stressed ending. The other important consideration is the morphophonological shape of the stem, including which letter or letters are stem-final, and whether the stressed prefix *vý-* is present. The relevant environments are listed in Table 2.24.

To begin, I will explain how the stressed imperative ending `S` is realized, since its rules are more straightforward. Examples are shown in Table 2.25.

The first example in Table 2.25, *pej*, comes from the five *j*-stem roots with a

Stem	Unstressed ending	Stressed ending
	U	S
CC or C′C	*i*	*í*
ŝ	*i*	*í*
g	–	*í*
j	–	–
C	'	*í*
-*vý*-C	*i*	n/a
V	*j*	*j*

Table 2.24: Realization of imperative endings

	péj 'drink!'	*spí* 'sleep!'	*stój* 'stand!'	*doí* 'milk!'
upper	pFéj<S	sp<S	stó<S	dó<í
lower	p_éj__	sp_í	stó_j	do_í

Table 2.25: Upper- and lower-side correspondences for imperatives with the stressed ending S

fleeting vowel discussed in the previous section.[32] Since these verbs are phonologically unique, their ending can be encoded as a two-level rule, instead of as an exceptional ending, as Zaliznjak suggests. The rule is simple, as shown in (38), which states that S is deleted when it is preceded by a *j*, followed by a deleted verbal stem boundary. The right context specifies possible sequences that can follow the base imperative inflection, and it will be discussed more below.

(38) `S:0 <=> j <:0 _ ( t e ) ( s [â:|'] )`

The second example in Table 2.25, *spi*, represents verbs with stems that end in any consonant other than *j*. Rule (39) declares the relations of such verbs. This rule states that S changes to *í* when it is preceded by any consonant other than *j* on the lower side, optionally followed by an upper-side M, followed by a deleted verbal stem boundary.

(39) `S:í <=> [:C - :j] (M:) <:0 _ ( t e ) ( s [â:|'] )`

The third example in Table 2.25, *stoj*, represents verbs with stems ending in a vowel. In these verbs, the imperative ending is a *j*. This change is defined in rule (40), which states that S changes to *j* when preceded by a lower-side vowel,

[32] Zaliznjak designates these verbs with code 11.

followed by a deleted verbal stem boundary.

(40) `S:j <=> :V <:0 _ ( t e ) ( s [â:|'] )`

The final example in Table 2.25, *doi*, represents a group of verbs whose imperative inflectional endings are encoded directly in `lexc`, because they do not follow the general rule.[33] This group is comprised of vowel-final stems of declension type `4b` or `4c`, and their imperative ending is the stressed *-í*.

In the following paragraphs, I explain the rules associated with the unstressed imperative ending `U`. The first examples are given in Table 2.26. The rule for changing `U` to *j*—as in the example of *duj*—is identical to rule (40) above. Likewise, rule (38) above also applies to deleting `U` in stems with *j*, as in the example of *vypej*.

	dúj 'blow!'	*výpej* 'drink!'	*lấg* 'lie down!'
upper	`dú<U`	`výpFej<U`	`lấg<U`
lower	`dú_j`	`výp_ej__`	`lấg__`

Table 2.26: Upper- and lower-side correspondences for imperatives with the unstressed ending `U`: *duj*, *vypej*, and *lâg*

The third example in Table 2.26 is interesting from a linguistic point of view. As will be shown later, the stem shape of *lâg* would typically result in a soft sign ending in the imperative: *lấg'*. However, Russian velars are such that they cannot be soft except before a front vowel, so the soft sign is not allowed in this context. This particular lexeme and its prefixed derivatives are the only velar stems with an unstressed imperative ending, so most descriptions of Russian treat it as an exception. However, especially because the difference is linguistically motivated, I prefered to encode this fact in a two-level rule, shown in (41), which states that `U` is deleted when preceded by *ấ*, followed by *g*, followed by a deleted verbal stem boundary.

(41) `U:0 <=> ấ g <:0 _ ( t e ) ( s [â:|'] )`

The remaining rules provide coverage for unstressed imperative endings on stems that end in consonant, of which there are four categories. An example of each category is given in Table 2.27.

[33] It would be possible to distinguish these stems from other vowel-letter stems phonologically by adding a *j* to the stems of other verbs, e.g. `stoj<S` instead of `sto<S`. However, this would complicate the whole verbal system too much to outweigh the addition of a handful of continuation classes.

	pómni 'remember!'	*mórŝi* 'wrinkle up!'	*výbegi* 'run out!'	*otvét'* 'answer!'
upper	pómn<U	mórŝ<U	výbeg<U	otvét<U
lower	pómn_i	mórŝ_i	výbeg_i	otvét_'

Table 2.27: Upper- and lower-side correspondences for imperatives with the unstressed ending U: *pómni*, *mórŝi*, *výbegi*, *otvét'*

Stems that end in two consecutive consonants—including both the sequence CC and C'C—have the imperative ending *-i*, as shown in the example *pomni*. Likewise, stems that end in the letter *ŝ* —which was historically a complex consonant /št/—also has the imperative ending *-i*, as shown in the example *morŝi*. These rules are defined in (42). Rule (42-a) covers forms like *pomni*, stating that U changes to *i* when it is preceded by a lower-side consonant, optionally followed by a lower-side soft sign, followed by any consonant other than *j* on the lower side, optionally followed by a deleted M, followed by a deleted verbal stem boundary. The consonant *j* is excluded here for two equally sufficient reasons. First, Russian orthography does not allow *j* immediately after another consonant. Second, including *j* would create a conflict with the U:0 version of rule (38) above.

(42) U:i <=>

a. :C (:') [:C-:j] (M:0) <:0 _ (t e) (s [â:|']))

b. :ŝ (M:0) <:0 _ (t e) (s [â:|'])

Rule (42-b) covers forms like *morŝi*, stating that U changes to *i* when it is preceded by a lower-side *ŝ*, optionally followed by a deleted M, followed by a deleted verbal stem boundary. Specifying the *ŝ* as lower-side is important here because it could be the result of a stem alternation, which would not have *ŝ* on the upper side.

The third example in Table 2.27, *vybegi*, represents a large group of perfective verbs with the stressed prefix *výˊ-*, whose stems end in a single consonant. One way of understanding the ending of these imperatives is by means of cascading phonological rules. Their base form, without the prefix *výˊ-*, would have a stressed imperative ending, and so their imperatives end in *-íˊ*, e.g. *begíˊ* 'run!'. Then, a later phonological process adds the prefix *výˊ-* which takes the stress away from the ending.

Identifying whether the string *vy* is a prefix or not is not trivial. If it is unstressed, then it can be safely ruled out. Likewise, if it does not occur at the beginning of the word, it might not be a prefix. However, there are instances of prefix stacking, in which the *vy* is buried under another prefix, namely *pere-* 're-', *po-*

'DISTRIBUTIVE', or *samo-* 'self-', as in *perevýberi* 're-choose!', *povýrubi* 'completely cut down a number of (trees)!', and *samovýrazis'* 'express yourself!'. There are also instances in which a word initial *vý-* is not a prefix, as in *výs'sâ* 'tower!'. This last group cannot be distinguished orthographically, so they must be distinguished by means of archiphonemes or special `lexc` continuation classes. The rule covering verbs such as *vybegi* is given in (43). This rule states that `U` changes to *i* when it is preceded by a sequence that optionally begins with either *po*, *pere*, or *samo*, followed by *v*, followed by *ý*, followed by a sequence of any number of any symbols, followed by either a) a vowel, followed by any consonant other than *g* or *j*, or b) any vowel other than *â* or *ấ*, followed by *g*, either of these optionally followed by a deleted `M`, followed by a deleted verbal stem boundary.

(43)
```
U:i <=> ( [ p o | p e r e | s a m o ] ) v ý ?*
[[:V [ :C - [ :g | :j ]]] | [[:V - [:ấ|:â]] :g]]
(M:0) <:0 _ ( t e ) ( s [â:|'])
```

Another class of verbs with the prefix *vý-* has the imperative inflection *-i*. This group is made up of vowel-final stems of declension class `4a` (stem-stressed imperative), such as *výdoi* 'milk dry!'. In essence, these verbs are exactly same as *doi* in Table 2.25, except that that the prefix *vý-* has pulled the stress to the stem. As in the case of *doi*, the imperative endings of verbs like *vydoi* are encoded explicitly in the `lexc` code without the help of two-level rules.

The last example in Table 2.27, *otvet'*, is the most canonical and straightforward, but because of the many exceptions already discussed, the two-level rule for expressing its relations must be written to avoid overlap. This rule is given in (44). The first part of the rule excludes stems with the prefix *vý-* and the second part of the rule excludes single-consonant stems with specific consonants that have been discussed previously. The rule states that `U` changes to *'* when it is preceded by any sequence of any symbols except for the sequence of *po* or *pere* or *samo* followed by *v* followed by *ý* itself followed by any sequence of any symbols, this entire sequence being followed by a lower-side vowel, followed by any lower-side consonant other than *ŝ*, *g* or *j*, optionally followed by a deleted `M`, followed by a deleted verbal stem boundary.

(44)
```
U:'<=> ?*-[([ p o | p e r e | s a m o ]) v ý ?*]
:V [:C - [:ŝ|:g|:j]] (M:0) <:0 _ ( t e )
( s [â:|'])
```

The only remaining piece of imperative inflection is the reflexive ending, which is realized as *-sâ* after consonants, and *s'* after vowels. Because the inflectional ending of imperatives can be realized as a vowel, consonant, or nothing, the reflexive

ending must be adjusted accordingly.[34] Examples of this are given in Table 2.28.

	molís' 'pray!'	*prošájsâ* 'farewell!'	*spráč'sâ* 'hide!'
upper	`mól<Ssâ`	`prošá<Usâ`	`sprâtM<Usâ`
lower	`mol_ís'`	`prošá_jsâ`	`spráč__'sâ`

Table 2.28: Upper- and lower-side correspondences for imperatives with the reflexive suffixes

By default, *-sâ* will remain *-sâ*, so the only context in which a change is necessary is when the imperative inflectional ending is realized as *-i* (or *í*), as in the example of *molís'* in Table 2.28. This is accomplished by rule (45), which simply states that *â* changes to *'* when it is preceded by a deleted verbal stem boundary, followed by *i* or *í*, followed by *s*.

(45) `<:0 [:i|:í] s _`

Verbal prefix fleeting vowels and voicing assimilation

There are a number of verbal prefixes in Russian that have fleeting vowels, but these vowels follow rules that are different from the type discussed in the nominal section above. The letters *o* and *ó* are the only prefixal fleeting vowels, and they occur in the following prefixes: *vo-*, *nado-*, *obo-*, *oto-*, *podo-*, *predo-*, *so-*, *vzo-*, *vozo-*, *izo-*, *nizo-*, and *razo-*. It would be impossible to reliably identify these prefixes based only on orthography, so their fleeting vowels were represented with archiphonemes to be certain that the applicable two-level rules do not operate in unwanted contexts. The stressed fleeting vowel is represented here as `O`, and the unstressed fleeting vowel is represented as `N`. By default, they are realized as *ó* and *o*, respectively. They are deleted, however, when followed by a single consonant followed by a vowel. Examples are given in Table 2.29.

	obgónit '(he) will outrun'	*razob'ët* '(he) will break it up'	*razbít'* 'to break up'
upper	`obOgón<it`	`razNbFéj<ët`	`razNbFéj<S`
lower	`ob_gón_it`	`razob__'_ët`	`raz_b_éj__`

Table 2.29: Upper- and lower-side correspondences for fleeting vowels in verbal prefixes

[34] Note that this adjustment only applies to the singular imperative. The plural imperative is itself suffixed with *-te*, which ends with a vowel, so the reflexive suffix is always realized as *-s'*. Therefore, the reflexive suffix of plural imperatives is already generated as such in `lexc`.

In the first example, *obgonit*, the fleeting vowel `O` is deleted because the fleeting vowel is followed by the sequence, *-gó-*. However, in the second example, *razob'ët*, the fleeting vowel `N` is not deleted because it is followed by the sequence *-b'-*. In the third example, *razbej*, the fleeting vowel is deleted because it is followed by the sequence *-bé-*. This rule is expressed in (46), which states that `O` and `N` are deleted when followed by a consonant, optionally followed by a deleted `F`, followed by a lower-side vowel.

(46) `[O:0|N:0] <=> _ C (F:0) :V`

A subset of the prefixes with fleeting vowels have one other change that is modeled using a two-level rule. The final *z* in *vz-*, *voz-*, *iz-*, *niz-*, and *raz-* changes to *s* when it is followed by an unvoiced consonant, as demonstrated in Table 2.30.

	razoš'ú́	*rassil*
	'(I) will rip/unseam'	'(he) ripped/unseamed'
upper	`razNšFéj<ú`	`razNší<l`
lower	`razoš__'_ú́`	`ras_ší_l`

Table 2.30: Upper- and lower-side correspondences for devoicing of *z* in verbal prefixes

The rule governing this change is given in (47), which states that *z* changes to *s* when it is preceded by *v*, *vo*, *i*, *ni*, or *ra*, and when it is followed by a deleted `O` or `N`, followed by an unvoiced consonant on the lower side, followed optionally by a deleted `F`, followed by a lower-side vowel.

(47)
```
z:s <=> [v|vo|vó:|i|í:|ni|ní:|ra|rá:]  _
[O:0|N:0] [:k|:p|:s|:t|:f|:x|:c|:č|:š|:ŝ] (F:0)
:V
```

Limitations of UDAR

The foundation of UDAR—as with almost all computational models of Russian—is Zaliznjak's *Grammatical dictionary of Russian*. Because of this, to the extent that UDAR is an accurate representation of the *Grammatical dictionary*, any limitations of the dictionary are inherited in the model. The most obvious example is missing lexemes. For example, the *Grammatical dictionary* contains no abbreviations or acronyms. It also has only a representative lexicon of proper nouns. Significant work is needed to improve coverage of these types of words.

Some paradigmatic insufficiencies have already been amended, such as the attenuated comparatives discussed above. Some other missing features have been identified, but have not yet been implemented. For example, productive affixes,

such as synthetic superlatives with the suffix *-ejšij* and/or the prefix *nai-*—as in *interesnejšij* 'the most interesting' or *naiprekrasnejšij* 'the very most wonderful'—are only modeled lexically in the *Grammatical Dictionary*, if at all. In order to expand the model to include these and other forms, some linguistic research is needed to ensure accurate implementations.

Also, the vocative form of personal names are not currently implemented in UDAR. Most vocatives can be generated using the same structure and rules as the genitive plural, but these forms do not always coincide; the vocative form does not have fleeting vowels, whereas the genitive plural does.

Similarly, as a more-or-less prescriptive enterprise, the *Grammatical dictionary* was not intended to model non-standard forms. This includes dialectal wordforms—such as the imperative *lâž'* 'lie down!' (compare Table 2.26 above)—as well as common mispellings and deliberate substandard language.[35] For applications that require analysis of non-standard Russian, UDAR will require expansion of both the lexicon and the rules.

There are a few systematic phenomena in Russian morphology that are not easily captured using a two-level morphology, especially affixation of various kinds. Some simple prefixation is easily performed using archiphonemes with a deleted trigger in relevant suffixes, as shown with attenuated comparatives in Table 2.17 above. However, more complicated cases of affixation are problematic, especially prefixation and suffixation in the verbal system.

One of the most prominent features of the Russian verbal system is aspectual pairs (or clusters), distinguished by prefixes and suffixes. For example, the simplex imperfective verb *smotret'* 'look.IPFV' is semantically paired with the perfective verb *posmotret'* 'look.PFV'. Some Russian part-of-speech taggers—including `mystem`— actually treat such pairs as a single lemma, so that *posmotret'* is seen as the perfective inflection of *smotret'*. Such an approach makes the assumption that these pairings can be established objectively (Janda, 2007). However, problematic cases are more a rule than an exception. For example, `mystem` analyzes the perfective verb *pročitaet* 'read.PFV-FUT-3SG' as an inflection of the verb *pročityvat'*, even though major Russian dictionaries list *čitat'* as the preferred counterpart. This example illustrates one of the most widespread causes of confusion about what constitutes a pair.

[35] For example, the so-called *âzyk padonkov* (a.k.a. *padonkaffskij* or *olbanskij jezyg*) has become commonplace in informal communication on the Internet. This practice distorts standard orthography, while still maintaining a phonetically viable representation of Russian. For example, the standard *privét* 'hi', is frequently rendered as *preved*, which, because of vowel reduction and word-final devoicing, should be pronounced the same.

2.3.4 Morphosyntactic tags

In this section, I define the morphosyntactic tags and other tags used in UDAR. Since UDAR is still under development, these tags will likely continue to evolve. Many tags are taken either directly or indirectly from the codes given in Zaliznjak (1977). For example, the code мо means "masculine, animate", which translates directly into corresponding morphosyntactic tags.

In addition, the tags used in UDAR are also influenced by conventions of the Giellatekno and Divvun research groups[36], as well as the Apertium open-source machine-translation project.[37]

The primary part-of-speech tags are given in Table 2.31. Although most part-of-speech categories may seem to be well-defined, there are instances that are more difficult to determine. For example, virtually all determiners—code мс-п in Zaliznjak—can be used substantively, i.e. as pronouns. In such cases, UDAR generates both readings.

Tag	Meaning
A	adjective
Abbr	abbreviation
Adv	adv
CC	coord. conjunction
CS	subord. conjunction
Det	determiner
Interj	interjection
N	noun
Num	numeral
Paren	parenthetical
Pcle	particle
Po	postposition
Pr	preposition
Pron	pronoun
V	verb

Table 2.31: Part-of-speech tags used in UDAR

Zaliznjak does not distinguish between types of conjunctions, using only one code, союз, for all conjunctions. However, there are obvious benefits to distinguishing between different kinds of conjunctions, especially when it comes to iden-

[36] giellatekno.uit.no and divvun.no
[37] apertium.org

tifying clause boundaries for syntactic parsing. For this reason, I have divided the conjunctions into two categories, coordinating and subordinating.

The particle category is used by many lexicographers of Russian as a miscellaneous category, with lexemes of wildly different syntactic behavior receiving the same tag. Efforts are being made to sort the particles into more meaningful categories (Endresen et al., 2016).

The sub-part-of-speech tags used in UDAR are given in Table 2.32. Most of these tags are not derived from Zaliznjak, but were deemed useful for morphosyntactic disambiguation and syntactic parsing. In other words, most of these tags are used to mark sets of words whose syntactic roles are somehow distinct from their general part-of-speech category. In some cases, creating these categories multiplies wordforms. For example, the pronoun *kto* 'who' can be used as an interrogative and a relativizer, much like English. I elected to generate two distinct readings for this lemma: `Pron Interr` and `Pron Rel`.

Tag	Meaning
All	'all'
Coll	collective (numeral)
Def	definite
Dem	demonstrative
Indef	indefinite
Interr	interrogative
Neg	negative
Pers	personal
Pos	possessive
Prcnt	percent
Prop	proper
Recip	reciprocal
Refl	reflexive
Rel	relative

Table 2.32: Sub-part-of-speech tags used in UDAR

The nominal morphosyntactic tags in UDAR are given in Table 2.33. Since Russian does not generally have gender in the plural, the `MFN` tag appears on plural modifiers to show that they can agree with nouns of any gender. The `MFN` tag also appears on a number of pluralia tantum, since there is no way to establish their gender in the singular.

The `Sem/Alt` tag is used for all inanimate proper nouns, such as place names, organizations and other entities. In the future, it may be beneficial to use more

specific tags.

The `Count` tag is used to distinguish plural genitive noun forms that are used with specific quantifiers: *desjat' čelovek* 'ten people' vs. *mnogo ljudej* 'lots of people'.

Tag	Meaning
Fem	feminine
Msc	masculine
Neu	neuter
MFN	any/unknown gender
Inan	inanimate
Anim	animate
AnIn	either animacy
Sem/Ant	first name
Sem/Pat	patronymic
Sem/Sur	surname
Sem/Alt	place names, organizations, etc.
Sg	singular
Pl	plural
Nom	nominative
Acc	accusative
Gen	genitive
Loc	locative (prepositional)
Dat	dative
Ins	instrumental
Loc2	locative2
Gen2	genitive2
Voc	vocative
Count	count
Ord	ordinal
Cmpar	comparative
Pred	predicative
Cmpnd	compound

Table 2.33: Nominal tags used in UDAR

The verbal morphosyntactic tags of UDAR are given in Table 2.34. One point worth discussing is the tags related to the deverbals: `PrsAct`, `PrsPss`, `PstAct` and `PstPss`. The verbal adverbs—such as *otkryvaja* 'while opening' or *vyigrav* 'having won'—are tagged with `PrsAct Adv` and `PstAct Adv`, respectively.

Participles—such as *polučivšij* 'which received'—are tagged with the appropriate deverbal tag combined with the appropriate nominal tags for adjectival inflection, e.g. `PstAct Msc AnIn Sg Nom`.

Tag	Meaning
Impf	imperfective
Perf	perfective
IV	intransitive
TV	transitive
Inf	infinitive
Imp	imperative
Pst	past
Prs	present
Fut	future
Sg1	1st person singular
Sg2	2nd person singular
Sg3	3rd person singular
Pl1	1st person plural
Pl2	2nd person plural
Pl3	3rd person plural
PrsAct	present active
PrsPss	present passive
PstAct	past active
PstPss	past passive
Pass	passive
Imprs	impersonal

Table 2.34: Verbal morphosyntactic tags used in UDAR

Finally, a set of tags is used to mark features that are not morphosyntactic, but regard points of usage, register, or sandhi. For example, the tags `Fac` and `Prb` tag wordforms as being facultative or problematic, as indicated in Zaliznjak (1977). Other tags, such as `Use/Obs` and `Use/Ant`, show that a form is obsolete or antiquated. The `PObj` tag is used for third-person personal pronouns with an epenthetic *n*, i.e. when they are the objects of prepositions: *ix* 'them.ACC' vs. *za nix* 'for them.ACC'. The `Leng` and `Elid` tags are used to mark forms that are longer or shorter than the canonical form, such as *novoj~novoju* 'new.FEM-SG-INS' or *novee~novej* 'new.COMP'.

The tag `Err/Sub` indicates that a form is nonstandard. This tag makes it possible to analyze nonstandard forms, such as *maxaeš'*, and by generating the same

lemma-tag sequence without the `Err/Sub` tag, one can convert it to a standard form, e.g. *mašeš'*.

2.3.5 Flavors of the FST

FSTs have closure properties which make it simple to filter and modify entries using regular expressions and operations such as union, subtraction and composition. XFST and HFST have functions built in to facilitate these operations. This makes it possible to easily create various "flavors" of UDAR, specific to a particular need.

For example, in order to generate unstressed wordforms, one option would be to generate the stressed wordform with the raw UDAR generator FST, and then remove the stress marks in post-processing. However, this approach increases the computational load of the generation process. A more efficient approach is to compose UDAR with an FST that converts stressed vowels to unstressed vowels. The resulting FST generates unstressed wordforms directly, without the need for post-processing. Likewise, the raw UDAR analyzer FST can be composed with an FST that makes stressed vowels optional. The resulting FST is maximally flexible, recognizing both stressed and unstressed wordforms.

Likewise, tags like `Err/Sub` and `Prb` mark nonstandard forms. By filtering out readings with such tags, it is possible to make FSTs that only analyze or generate normative wordforms.

2.4 Evaluation

In order to evaluate the coverage and speed of UDAR, I compared it with other morphological transducers and related resources that were readily available: mystem3, pymorphy2, and the OpenCorpora lexicon, which is the source of pymorphy2's lexicon. We can expect UDAR's performance to be weaker than mystem3 and pymorphy2 for multiple reasons. Yandex is a large technology company, and has likely put significant resources into developing mystem3, one of its foundational NLP technologies. Likewise, pymorphy2 has benefited significantly from the OpenCorpora project, which has led to an expanded lexicon with more coverage than Zaliznjak's original dictionary. However, UDAR occupies a unique position as a Russian transducer that handles stress, and the following tests demonstrate that its performance is at least comparable to these other resources.

In general, one of the most difficult parts of comparing and evaluating rule-based part-of-speech taggers is that there is no standard for tag names, categories, or order, so establishing equivalence of tag strings is not always straightforward. This is certainly true of Russian part-of-speech taggers. Since all of the prominent existing morphological engines are based on Zaliznjak's grammatical dictionary,

	mystem3	UDAR
tokens missing	17 721 987	34 853 902
tokens tagged	236 348 025	295 962 831
tot tokens	254 070 012	330 816 733
% tagged	93.0%	89.5%

Table 2.35: Coverage of wikipedia lexicon by UDAR and mystem3.

it is reasonable to assume that the paradigms of the analysers will be equivalent, even if the particulars of their tagging conventions differ. In fact, an informal comparison of their outputs indicates that they are effectively the same. Because of this, no attempt was made to translate between the tagsets of each analyzer.

2.4.1 Coverage

I employed two different approaches to evaluate UDAR's coverage. First, I directly compared UDAR's lexicon with the lexicon of the OpenCorpora project. This was achieved by using UDAR to analyze each surface form of OpenCorpora's lexicon, and counting the number of forms for which no analysis was returned. OpenCorpora has 4 909 850 wordforms in its lexicon and 3 394 466 are recognized by UDAR, which is a coverage of 69%. As a reference point, mystem3 recognizes 4 768 520 wordforms from the OpenCorpora lexicon, which is 97% coverage. Unfortunately, because mystem is not open-source, it is not possible to test the coverage of UDAR and pymorphy2 on mystem3's lexicon. Even so, these results indicate that UDAR's lexicon is much smaller than both pymorphy2 and mystem3.

A second, more practical method for testing lexicon coverage is on unrestricted text. This was achieved by analyzing a dump of the Russian wikipedia and counting the number of unknown tokens in the output. The results of this test are shown in Table 2.35. Notice that the tokenization of mystem3 and UDAR are significantly different. This is primarily due to the fact that UDAR treats punctuation marks as tokens. On this test, UDAR covered 89.5% of the corpus and mystem3 covered 93.0%.

2.4.2 Speed

For some applications, the speed of the morphological analyzer is important, especially in live, interactive applications. The speed of UDAR was compared with other taggers by processing the surface forms of the OpenCorpora lexicon. Results are given in Table 2.36.

	time (sec.)	rate (words/sec.)
UDAR (xfst)	92.507	59 290
UDAR (hfst)	94.878	57 808
mystem3	66.809	82 095
pymorphy2	1310.808	4 184

Table 2.36: Speed comparison processing the OpenCorpora lexicon list (5 484 696 tokens)

Both UDAR and mystem3 perform on an order of magnitude faster than pymorphy2. However, most likely there are ways to trim overhead from pymorphy2's processes to increase its speed. In any case, the results in Table 2.36 show that UDAR is a very fast transducer, though not the fastest. This makes UDAR suitable for interactive applications that require processing large amounts of data in short time spans.

2.5 Potential applications

The previous section has shown that UDAR is a very fast analyzer, even if it is not the fastest. UDAR also achieves a reasonably high coverage of wordforms found in the wikipedia corpus. This makes it a good alternative to other existing taggers, especially if there is a need for processing stressed wordforms. In this section, I outline a few possible applications for which UDAR is particularly well-suited.

It should be noted that UDAR is being developed in conjunction with a constraint grammar in the tradition of Karlsson (1990), which removes readings that are not warranted by the syntactic context. This constraint grammar is continually under development and is already removing 43% of extra readings from UDAR's output (Tyers and Reynolds, 2015). When it comes to disambiguation of wordforms that have marked stress, UDAR has the added benefit that it does not output readings that contradict that stress position in the input. For example, the surface form *sestry* 'sister' is ambiguous between SG-GEN (*sestrý*) and PL-NOM (*sëstry*), and pymorphy2, mystem3 and UDAR all output both readings. But with the input *sestrý*—which is unambiguously SG-GEN—mystem3 outputs the same two readings and pymorphy2 treats it as an unknown token, whereas UDAR outputs only the SG-GEN reading. This type of ambiguity can be found in about 7.5% of tokens in running text (Reynolds and Tyers, 2015).

Stress annotation

UDAR can be used to mark stress in running text. This is achieved by piping the results of the UDAR analysis transducer into the UDAR generator transducer,

along with a decision heuristic for wordforms with more than one possible stress position. A recent study found that in combination with the constraint grammar mentioned above, UDAR achieves up to 96.15% accuracy (Reynolds and Tyers, 2015).[38]

Computer-Assisted Language Learning

Another application for a stress-aware morphological engine is computer-assisted language learning, especially in interactive applications. For example, Reynolds et al. (2014) implemented an intelligent web app and Firefox extension to allow learners to transform webpages into interactive grammar workbooks on various grammatical topics, such as word stress, verb conjugation, noun declension, etc. In that application, UDAR and its companion constraint grammar are used to analyze the text on a webpage. This information is used to select relevant tokens for grammatical activities. In addition, one of the available activities for each topic is multiple-choice. For this activity, the learner needs alternatives to the original form in the text, and UDAR is used to intelligently generate distractor wordforms from the same lemma. In this context, UDAR is used both to analyze and to generate Russian wordforms for the learner.

Spell-checking

It is possible to create spellcheckers from two-level morphologies, such as UDAR. And since UDAR explicitly models stress position of each wordform, this leads to the possibility of stress-sensitive spellcheckers. Such a spellchecker could be used by language teachers to check texts that they have prepared for their students. It could also be used by learners to help write texts with marked stress.

Machine translation

Finally, because UDAR was built with tags compatible to the Apertium project, UDAR can easily become a component in a machine translation pipeline. As a major world language, resources already exist for machine translation of Russian to other major languages, but Russian serves as a regional Lingua Franca for many minority languages, including many of the Turkic languages already included in the Apertium project. An open-source morphological analyzer such as UDAR could therefore help facilitate the use of Russian as a pivot language for minority languages of the region.

[38] Accuracy of 93.21% was achieved with less than 0.5% error. Higher accuracies were achieved by using guessing algorithms of various kinds, with higher error rates.

2.6 Conclusions and future work

This chapter has introduced UDAR, the only large-scale free and open-source Russian morphological engine designed to process stressed wordforms. I outlined some of the strategies employed in the `lexc/twolc` source code to model Russian morphology/orthography. UDAR's performance was evaluated against other existing Russian morphological engines. Although UDAR does not outperform mystem3's coverage or speed, it does have reasonably good lexicon coverage and very fast speed. Its performance is very adequate for modern applications that deal with stressed wordforms.

Future work will improve the coverage of UDAR, especially by importing lexemes from OpenCorpora's lexicon. In addition, steps will be taken to implement guessing algorithms for unknown wordforms, such as `hfst-guess`.

Another possibility that has not been widely explored in computational morphology is the use of weighted transducers. We intend to experiment with adding weights to UDAR to improve its accuracy.

Finally, since UDAR occupies a specialized place as a morphological engine for language learners, an obvious extension of UDAR is learner-language analysis. Implementing a learner error taxonomy, such as in Dickinson (2010), to the `lexc` source code of UDAR would allow for compiling a learner-language version of UDAR. With a learner-language flavor of UDAR in place, it would also become possible to generate a learner-language spell-checker, as discussed in Section 2.5 above.

Chapter 3

Morphosyntactic disambiguation and dependency annotation[1]

This chapter presents preliminary work on a constraint-grammar based disambiguator for Russian. Russian is a Slavic language with a high degree of both in-category and out-category homonymy in the inflectional system. The pipeline consists of the finite-state morphological analyzer introduced in Chapter 2 and the constraint grammar described in the present chapter. The constraint grammar is tuned to be high recall (over 0.99) at the expense of lower precision.

3.1 Introduction

This chapter presents a preliminary constraint grammar for Russian. The main objective of this research is to produce a high-recall grammar to serve as input into other natural language processing tasks. *High recall* means that the grammar is intended to remove only those readings that can be ruled out with high confidence, leaving ambiguity that is difficult or impossible to resolve based on morphosyntactic and syntactic context. For example, the simple sentence in example (1), has two tokens with ambiguous readings.

(1) *Ona sxodila v park.*
She went.IPFV/PFV to park.SG-NOM/ACC.
'She went to the park'

[1]The research presented in this chapter was carried out in collaboration with Francis Tyers. It is a modified version of Tyers and Reynolds (2015). Many of the constraint grammar rules and much of the experimental methodology were conceptualized cooperatively, but Dr. Tyers deserves full credit for the implementation of the actual constraint grammar rules.

The token *sxodila* can be imperfective (meaning 'go down'), or perfective (meaning 'make a quick trip there and back'). The question of which reading to select cannot be resolved without understanding the broader narrative, so our grammar does not attempt to resolve this ambiguity. On the other hand, the token *park* has two readings as output from the morphological analyzer: SG-NOM or SG-ACC. The preposition *v* 'in/to' can govern ACC or LOC, but not NOM,[2] so the SG-NOM reading can be removed with high confidence.

There are two reasons to maintain high recall. First, one of the primary applications for this constraint grammar is computer-assisted language learning. In this domain, erroneous analyses can lead to significant frustration for learners. So with regard to disambiguation, developing intelligent computer-assisted language learning applications requires a kind of intuitionistic or epistemic logic where the developer can base control flow operations not just on the most probable readings, but on the relative certainty of those readings. For instance, if a program that automatically generates grammatical exercises were to process example (1), the reliability of the program's output would depend on the nature of its morphological analysis input. A traditional part-of-speech tagger would output a single reading for each token. If the tagger happened to guess wrong on the aspect of *sxodila*, then the language-learning program would blindly make erroneous exercises based on faulty input. On the other hand, if the program receives input that maintains the ambiguity, then the program "knows whether it knows", and it can avoid generating exercises on tokens where the relevant tags are uncertain. This kind of logic is only possible with a tagger that is capable of giving ambiguous output.

A second reason for maintaining high recall in our constraint grammar is that it is frequently the case that competing readings can be distinguished only by considering idiosyncratic collocational information. Writing rules to capture each of these idiosyncrasies would be very complicated and time-consuming. For such cases, we expect that using a voting setup, in which a probabilistic model chooses between readings that remain after our rules are applied, would be both both more effective and simpler to implement than a host of low-coverage rules. A high-recall constraint grammar follows the maxim "Don't guess if you know", after which a a probabilistic model can be used to effectively capture most-likely tags based on usage data.

The chapter is laid out as follows: Section 3.2 presents a review of related work in Russian morphological disambiguation; Section 3.3 gives an overview of ambiguity in Russian; Section 3.4 describes our analysis pipeline; Section 3.5 gives

[2]In fact, the preposition *v* can govern NOM, in phrases such as *pojti v professora* 'become a professor; *lit.* go among professors', but this use of the preposition is limited to plural animate nouns, so the word *park* cannot be in NOM in the example discussed above. Some scholars prefer to call this Acc2, as is done in the Russian National Corpus.

an account of our development process; Section 3.6 presents an evaluation of the system; and 3.7 presents conclusions, as well as an outlook for continuing research and development of our grammar.

3.2 Related work

As explained in Chapter 2, state-of-the-art morphological analysis in Russian is primarily based on finite-state technology (Nozhov, 2003; Segalovich, 2003).[3] Almost without exception, all large-scale morphological analyzers of Russian are ultimately based on the *Grammatical Dictionary of Russian* (Zaliznjak, 1977). This dictionary gives fine-grained morphological specifications for more than 100 000 words, including inflectional endings, morphophonemic alternations, stress patterns, exceptions, and idiosyncratic collocations. The morphological analyzer used in the present chapter is also based on Zaliznjak's dictionary.[4] This finite-state transducer (FST) generates all possible morphosyntactic readings of each wordform, regardless of their frequency or probability. Because Russian is a relatively highly inflected language, broad coverage is important, but widespread homonymy leads to the generation of many spurious readings, as discussed in Section 3.3 below. Because of this, one of the foundational steps in Russian natural language processing is homonym disambiguation.[5]

Two main approaches to Russian have been discussed by researchers: rule-based and probability-based. Rule-based approaches to Russian have a long tradition. For instance, the well-known ETAP system (Cinman and Sizov, 2000, e.g.)—which has been in development since the 1960s—generates all possible syntactic dependencies, after which infelicitous trees are removed on the basis of semantics, case goverment, and/or morphosyntactic agreement. This process is repeated iteratively, until only globally felicitous trees remain. Similarly, the Synan system—which is part of the DiaLing project published at `http://www.aot.ru`—can perform morphosyntactic disambiguation indirectly by means of syntactic rules, not including semantics, case goverment, or long-distance dependencies. Synan successfully removes approximately 30% of lexical and morphosyntactic ambiguity in the input text (Sokirko and Toldova, 2004).

[3] Machine-learning approaches have also been successfully applied to Russian, most notably by Sharoff et al. (2008a).

[4] Our transducer is implemented using a two-level morphology (Koskenniemi, 1984), and can be compiled using either `xfst` (Beesley and Karttunen, 2003) or `hfst` (Linden et al., 2011). For more information, please refer to Chapter 2.

[5] In the Russian-language research literature, this is referred to as *snjatije omonimii* 'removal of homonymy' or *snjatie leksičeskoj i morfologičeskoj neodnoznačnosti* 'removal of lexical and morphological ambiguity'.

The goal of systems such as ETAP and Synan is to give a syntactic analysis, and their processing speed is strongly tied to the degree of morphosyntactic and lexical ambiguity in the input. Disambiguation—which is not the primary purpose of these systems—is performed very slowly, and at least in the case of Synan, not completely.

The other overall approach to lexical and morphosyntactic disambiguation is the use of probabilistic models which output only a single most likely reading. Popular probabilistic methods have been applied to Russian, such as Hidden Markov Models (Sokirko and Toldova, 2004; Sharoff et al., 2008a), but other approaches have been developed specifically for Russian as well. For instance, Zelenkov et al. (2005) relies on estimates for the "difficulty of choosing the lemma" which is based on hypothesized distances between parts of speech and distances between certain morphosyntactic values, such as case.

Probabilistic approaches have been reported to have high accuracy: Sokirko and Toldova (2004) report 97.8%; Sharoff et al. (2008a) report 95.28%; and Zelenkov et al. (2005) report 97.42%. Although these results are relatively high, probabilistic models cannot maintain the intuitionistic/epistemic logic that we need for our target application of intelligent computer-assisted language learning. In other words, probabilistic models do not "know whether they know", and cannot allow the application designer to keep track of what the system can deduce with near-certainty.

Constraint grammar (CG) is a paradigm of natural language processing in which linguist-written context-dependent rules are compiled to systematically remove readings from running text (Karlsson, 1990; Karlsson et al., 1995). Constraint grammars can process text faster than ETAP and Synan, and because they are rule-based, they can be tuned to high recall to allow for epistemic logic in designing language-learning applications. The CG paradigm has been successfully applied to languages from many different language families: Basque, Breton, Catalan, Croatian, Danish, English, Esperanto, Faroese, Finnish, French, German, Greenlandic, Irish, Irish Gaelic, Italian, Komi, Norwegian (Bokmål and Nynorsk), Portuguese, Lule Sami, North Sami, Spanish, Swedish, Swahili, and Welsh. To our knowledge, the only CG developed for a Slavic language is the Croatian CG developed by Peradin and Šnajder (2012), which achieves 88% precision and 98% recall for morphosyntactic analysis.

3.3 Ambiguity in Russian

Different kinds of ambiguity are resolved by different means. There are two major kinds of ambiguity that are relevant to detailed part-of-speech tagging. *Lexical*

ambiguity refers to instances where a given token can belong to more than one lexeme. *Morphosyntactic* ambiguity occurs in tokens that have more than one possible set of morphosyntactic tags. These two categories of ambiguity are not mutually exclusive, and consequently, we find three different categories of ambiguity: intraparadigmatic ambiguity (purely morphsyntactic), morphosyntactically congruent lexical ambiguity (purely lexical), and morphosyntactically incongruent lexical ambiguity (both morphosyntactic and lexical). The following examples make use of word stress ambiguity to illustrate each kind of underlying ambiguity, but it should be noted that most lexical and morphosyntactic ambiguity does not result in stress position variation.[6] *Intraparadigmatic* ambiguity refers to homographic wordforms belonging to the same lexeme, as shown in (2).

(2) Intraparadigmatic homographs
   a. *téla* 'body.SG-GEN'
   b. *telá* 'body.PL-NOM'

The remaining two types of ambiguity occur between lexemes. *Morphosyntactically incongruent* lexical ambiguity occurs between homographs that belong to separate lexemes, and whose morphosyntactic values are different, as shown in (3).

(3) Morphosyntactically incongruent homographs
   a. *nášej* 'our.F-SG-GEN/DAT/LOC/INS'
      našéj 'sew on.IMP-2SG'
   b. *dorógа* 'road.N-F-SG-NOM'
      dorogá 'dear.ADJ-F-SG-PRED'

Morphosyntactically congruent lexical ambiguity occurs between homographs that belong to separate lexemes, and whose morphosyntactic values are identical, as shown in (4).

(4) Morphosyntactically congruent homographs
   a. *zámok* 'castle.SG-NOM'
      zamók 'lock.SG-NOM'
   b. *zámkov* 'castle.PL-GEN'
      zamkóv 'lock.PL-GEN'
   c. etc. ...

[6]Written standard Russian does not typically indicate stress position, but knowing stress position is essential for pronunciation. A recent study by Reynolds and Tyers (2015) found that about 7.5% of tokens with stress ambiguity could have their stress position resolved indirectly by resolving lexical and/or morphosyntactic ambiguity.

Note that a given token can exhibit more than one kind of ambiguity. For example, the wordform *zamkov* has the readings given in (5). The ambiguity between (5-a) and (5-b) is morphosyntactically congruent, and the ambiguity between (5-a)/(5-b) and (5-c) is morphosyntactically incongruent.

(5) a. *zamok*[1].N-MSC-INAN-PL-GEN
b. *zamok*[2].N-MSC-INAN-PL-GEN
c. *zamkovyj*.ADJ-MSC-SG-PRED

Table 3.1 shows the prevalence of each kind of ambiguity in the corpus of Russian presented in section 3.6.1 below. The first column shows the proportion of all tokens that have each kind of ambiguity. The second column shows what proportion of ambiguous tokens exhibit each kind of ambiguity. Note that these proportions do not sum to 100%, since a given token may exhibit more than one kind of ambiguity.

Type	**all tokens**	**ambiguous tokens**
Intraparadigmatic	59.0%	90.9%
MS-incongruent lexical	27.7%	42.7%
MS-congruent lexical	1.2%	1.8%

Table 3.1: Frequency of different types of morphosyntactic ambiguity in unrestricted text

These data show that almost 59% of all tokens in Russian running text exhibit intraparadigmatic ambiguity, and 27.7% of all tokens exhibit lexical ambiguity that is morphosyntactically incongruent. These results show that most morphosyntactic ambiguity in unrestricted Russian text is rooted in intraparadigmatic and morphosyntactically incongruent lexical ambiguity. Detailed part-of-speech tagging with morphosyntactic analysis can help disambiguate these forms. On the other hand, morphosyntactically congruent lexical ambiguity represents only a very small percentage of ambiguous wordforms, and it can be resolved most readily by means of word sense disambiguation, as opposed to detailed part-of-speech tagging. Because of this difference, we leave morphosyntactically congruent lexical ambiguity to future work.

3.4 Analysis pipeline

In the following section, I give a brief description of the morphological analyzer on which our constraint grammar is based. For a more complete description, please refer to Chapter 2.

3.4.1 Morphological analyzer

The morphological transducer used in this study is primarily based on Zaliznjak's *Grammatical dictionary of Russian* (Zaliznjak, 1977), including the 2001 version's appendix of proper nouns. It also includes neologisms from Grishina and Lyashevskaya's *Grammatical dictionary of new Russian words* (Grishina and Lyashevskaya, 2008), which is intended to be a supplement to Zaliznjak's dictionary with words found in the Russian National Corpus.[7] Example (6) gives some examples of the FST's output.

(6) a. *konečnyj*<adj><m><nn><sg><nom>
'finite'
b. *avtomat*<n><m><nn><sg><nom>
'automaton, sub-machine gun'

3.4.2 Disambiguation rules

Our grammar is implemented using the vislcg3 constraint grammar parser (`http://beta.visl.sdu.dk/cg3.html`). The constraint grammar is composed of 299 rules which are divided into four categories: Safe, Safe heuristic, Heuristic, and Syntax labeling. The distribution of rules is shown in Table 3.2, and each category is discussed below.

	SELECT	REMOVE	MAP
Safe	16	34	–
Safe heuristic	89	76	–
Heuristic	26	52	–
Syntax labeling	–	–	6

Table 3.2: The distribution of rules in reliability categories and syntactic role labeling.

Safe rules should represent real constraints in the language. Examples might be that a preposition cannot directly precede a finite verb or that a noun cannot be in the prepositional case without a preceding preposition. If exceptions to safe rules can be found, they should be quite rare, as well as being unacceptable or strange to some native speakers. In order to be considered safe, a rule must fire correctly 100% of the time in our development sample, as described in Section 3.5 below.

Safe heuristic rules should deal with highly frequent tendencies in the language. In order for a rule to be considered a safe heuristic, it must fire correctly at least 98% of the time in our development sample. For example: remove a genitive

[7] `http://dict.ruslang.ru/gram.php`

reading of a noun/adjective/pronoun at the beginning of a sentence if the word is capitalized and there is no verb governing the genitive found to the right and there is also no negated verb to the right. This rule relies on the fact that if the genitive is in first position in the sentence it cannot modify anything before it, and no preposition can be governing it. This kind of rule often relies on completeness of sets, in this case the set of verbs that can take a genitive complement.

Heuristic rules are those which we do not consider linguistic constraints, but express ad-hoc preferences, often dealing with extremely rare readings on high-frequency tokens. For example: remove the verbal adverb reading of *takaja*, which could be the feminine singular nominative of *takoj* 'such' or the verbal adverb of *takat'* 'express agreement by saying *tak*'. Heuristic rules are, for the most part, a matter of practical engineering, as opposed to language modeling. Native Russian speakers can usually formulate counterexamples to the Heuristic rules with ease, but in actual usage such counterexamples should be relatively rare. Given a large hand-annotated corpus we believe that most of the heuristic rules would be better replaced with information learned from the corpus through stochastic methods.

Finally, the syntactic role-labeling rules are prerequisite for generating a dependency parse of a sentence. For example, the label for objects of prepositions `@P<-` is added to readings of tokens that come immediately after prepositions. So far, only a few labels have been assigned, and the rules have not been well developed. In this study we do not evaluate the performance of our grammar's rules for syntactic role labeling.

3.5 Development process

A common approach taken when writing constraint grammar rules is to apply the existing rule set to a new text, write new rules to deal with the ambiguities, then apply the rules to a hand-annotated corpus to see how often the rule disambiguated correctly (Voutilainen, 2004).

Due to the lack of a hand-annotated corpus compatible with our morphological analyzer, we adopted a slightly modified technique. We picked random texts from the Russian Wikipedia,[8] ran them through the morphological analyzer, wrote rules, and then ran the rules on the whole Wikipedia corpus. For each rule, we collected around 100 example applications and checked them. If a rule selected the appropriate reading in all cases, we included it in the *safe* rule set, if it removed a valid reading in less than three cases, then we included it in the *safe heuristic* rule set.

[8]The Russian Wikipedia was chosen as a testing corpus as it is the largest, freely licensed corpus of Russian available on the internet. Encyclopedic text is, of course, not representative of all genres of Russian text.

Otherwise we either discarded the rule or included it in the heuristic rule set.

3.6 Evaluation

In order to demonstrate how our evaluation metrics are calculated, an example sentence is given in Figure 3.1: "В ноябре 1994 года в Танзании начал работу Международный трибунал по Руанде." 'The work of the International Tribunal for Rwanda started in Tanzania in November 1994.' Numbers given after `SELECT:` or `REMOVE:` refer to the source code line numbers of rules according to which the action was taken. The terms `SELECT` and `REMOVE` refer to the type of rule; a semicolon at the beginning of the line indicates that a reading is removed by the CG. A `SELECT` rule removes all but the selected reading(s). Each of the rules which are relevant to Figure 3.1 are presented in Figure 3.2 with English translations.

In this sentence, there are 13 tokens (including punctuation) and 23 readings, so the input ambiguity before running the constraint grammar is 1.77 readings per word. The constraint grammar removes 5 readings (indicated by a semicolon at the beginning of the line), so the output ambiguity is 1.38 readings per word.

Precision, recall and accuracy are computed on the basis of four types of outcome: true positive, false positive, true negative, and false negative. Here *positive* means that a reading is not removed by the CG, and *negative* means that a reading is removed by the CG. Every reading in the CG output that is in the gold-standard reading(s) for that token is a *true positive* (TP), and every reading in the CG output that is not in the gold-standard reading(s) for that token is a *false positive* (FP). Similarly, every reading that is removed by the CG and is not found in the gold-standard reading(s) for that token is a *true negative* (TN), and every reading that is removed by the CG and *is* found in the gold-standard reading(s) for that token is a *false negative* (FN).

Precision is computed as $\frac{TP}{TP+FP}$, which yields the percentage of readings remaining in the CG output that are found in the gold standard. The precision score penalizes the failure to remove incorrect readings. In Figure 3.1, the CG output contains 18 readings, 13 of which are correct, so precision is 0.72. Recall is computed as $\frac{TP}{TP+FN}$, which yields the percentage of readings in the gold standard that remain in the CG output. The recall score penalizes removing readings that are correct. In Figure 3.1, the CG contains the correct reading for all 13 tokens, so recall is 1.0. Accuracy is computed as $\frac{TP+TN}{TP+FP+TN+FN}$, which yields the percentage of readings in the morphological analyzer's output that were treated correctly by the CG. In Figure 3.1, of 23 original readings, 13 readings were retained correctly, and 5 readings were removed correctly, so the accuracy is $\frac{13+5}{23} = 0.78$.

```
 "<В>"
    "в" pr

"<ноябре>"
    "ноябрь" n m nn sg prp

"<1994>"
    "1994" num

"<года>"
    "год" n m nn sg gen SELECT:r462
;   "год" n m nn pl nom fac SELECT:r462

"<в>"
    "в" pr

"<Танзании>"
    "Танзания" np al f nn pl acc
    "Танзания" np al f nn sg prp
;   "Танзания" np al f nn pl nom REMOVE:r424
;   "Танзания" np al f nn sg dat REMOVE:r433
;   "Танзания" np al f nn sg gen REMOVE:r433

"<начал>"
    "начало" n nt nn pl gen
    "начать" vblex perf tv past m sg
;   "начать" vblex perf iv past m sg REMOVE:r769

"<работу>"
    "работа" n f nn sg acc

"<Международный>"
    "международный" adj m an sg nom
    "международный" adj m nn sg acc

"<трибунал>"
    "трибунал" n m nn sg acc
    "трибунал" n m nn sg nom

"<по>"
    "по" pr

"<Руанде>"
    "Руанда" np al f nn sg prp
    "Руанда" np al f nn sg dat

"<.>"
    "." sent
```

Figure 3.1: Example output from the morphological analyzer and constraint grammar

Safe:

```
(462) SELECT Gen IF (0 Year) (-1 Num LINK -1 Months LINK -1
Pr/V);
```

- Select genitive reading of 'года' if there is a numeral immediately to the left, before that there is a month and before that there is the preposition 'в'.

```
(424) REMOVE Nom IF (-1C Pr) ;
```

- Remove nominative case if there is a word which can only be a preposition immediately to the left.

```
(433) REMOVE NGDAIP - Acc - Prp - Loc IF (-1C* Pr/V OR Pr/Na
BARRIER (*) - Adv - Comp - DetIndecl - ModAcc - ModPrp);
```

- Remove all cases apart from accusative, prepositional and locative if 'в' or 'на' are found to the left and are unambiguous. The barrier is anything that cannot be found inside a noun phrase.

Safe heuristic:

```
(769) REMOVE IV IF (0 TV OR IV) (1C Acc) (NOT 1 AccAdv);
```

- Remove an intransitive reading of a verb if the next word can only be accusative and is not in the set of nouns which can be used adverbially in accusative.

Figure 3.2: Constraint grammar rules relevant to Figure 3.1

Domain	Tokens	Precision	Recall	F-score	Ambig. solved
Wikipedia	7,857	0.506	0.996	0.671	44.92%
Literature	1,652	0.473	0.984	0.638	42.95%
News	642	0.471	0.990	0.638	41.60%
Average	10,150	0.498	0.994	0.663	44.39%

Table 3.3: Results for the test corpora

3.6.1 Corpus

In order to evaluate the grammar we hand-annotated 10,150 words of Russian text from Wikipedia articles, public domain literature and freely-available news sources. The annotated texts are available online under the CC-BY-SA licence.[9] We chose to annotate our own texts as opposed to using a well-known hand-annotated corpus such as the Russian National Corpus (RNC) for two main reasons: the first was that the RNC is not freely available; the second was that the standards for tokenization, part-of-speech and morphological description are different from our morphological analyzer.

Hand-annotation proceeded as follows: The text was first passed through the morphological analyzer, and then a human annotator read through the output of the morphological analyzer, removing readings which were not appropriate in the context. This annotated text was then checked by a second annotator.

Table 3.3 gives a quantitative evaluation of the performance of our CG on the test corpus. The F-score is a harmonic average of precision and recall.

3.6.2 Qualitative evaluation

In this section, I give a qualitative evaluation of errors made by the CG. The errors reported below were sorted into categories for the sake of discussion.

Incomplete linguistics In some cases a rule did not take into account grammatical possibilities in the language. Consider, for example, the simple rule and counterexample given in (7). The rule in (7-a) removes the determiner reading if the token is followed immediately by the word *ne* 'not'. However, example (7-b) shows a postposed determiner followed by *ne*. The sentence is perfectly grammatical, but the rule does not take into account the flexible word order that is possible in Russian. and (8).

[9] `https://svn.code.sf.net/p/apertium/svn/languages/apertium-rus/texts/`

(7) a. `REMOVE Det IF (0 Det OR Pron) (1C Ne) ;`
b. *. . . a možet byt' i ran'še, i fakt* ***ètot*** *ne raz poražal menja . . .*
‘ . . . and maybe even earlier, and **this** fact surprised me several times . . . ’

Another example of a rule that was badly conceived from a linguistic point of view is given in (8-a). Here the rule removes the determiner reading if followed by a comma and a conjunction. However, this rules does not take into account the possibility of interposed parentheticals, such as the one in example (8-b).

(8) a. `REMOVE Det IF (0 Det OR Pron) (1 Cm LINK 1 CC OR CS) ;`
b. *No kakie,* ***odnako*** *že, dva raznye sozdanija, točno obe s dvux raznyx planet!*
‘But what, **exactly** , two different creatures, truly both from two different planets!’

Bad linguistics In some cases a rule was simply incorrectly specified. For example, the rule in (9-a) was designed to solve the ambiguity between short-form neuter adjectives and adverbs. However there is no reason why we should prefer an adverb over an adjective after an adverb, as in example (9-b). The rule’s context was too broad, so the corrected rule should have the `Adv` removed from position `-1C`.

(9) a. `REMOVE A + Short IF (-1C Fin OR Adv OR A) (0C Short OR Adv) ;`
b. *. . . Potomu čto sovsem* ***neprijatno*** *prosnut'sja v grobu pod zemleju.*
‘. . . because (it is) really **unpleasant** to wake up in a coffin under the ground.’

Incomplete barrier Some rules suffered from incomplete barriers, which is something that would benefit from a more systematic treatment. For example, the rule in (10-a) removes the nominative reading of *èlektrificirovannye* because the preposition *v* is found in the preceding context. The barrier for the rule can be updated to capture tokens such as *1960-x*.

(10) a. `REMOVE NGDAIP - Acc - Prp - Loc IF (-1C* Pr/V OR Pr/Na BARRIER (*) - Adv - Comp - DetIndecl - ModAcc - ModPrp) ;`
b. *V 1960-x* ***èlektrificirovannye*** *vysokoskorostnye železnye dorogi pojavilis' v Japonii i nekotoryx drugix stranax.*

'In the 1960's **electrified** high-speed railways appeared in Japan and some other countries.'

Incomplete sets In some cases the rule was a good generalization, but made use of a set which was incomplete. An example of this is given in (11-a), which removes dative case readings for several reasons, one of them being that there is no verb that governs the dative case in the sentence.

(11) a. `REMOVE Dat IF (NOT 0 Prn/Sebe) (NOT 0 Anim OR Cog OR Ant) (NOT 0 Pron) (NOT 1* V/Dat) (NOT -1* V/Dat) (NOT -1* Prep/Dat) (NOT -1C A + Dat) ;`

b. *V svjazi s ètim **ortodoksal'nosti** stali protivopostavljat' eres'.*
'In connection with this, heresy began to be seen in opposition to **orthodoxy**.'

Here the rule fails for the simple reason that the set `V/Dat` does not contain the verb *protivopostavljat'* 'opposed to' which takes a dative argument. A more complete set would lead to correct results.[10]

Rule interaction Some rules failed because of unexpected interactions with other rules. The rule in (12-b) removes transitive readings if there are no accusatives in the sentence. The strong accusative rule in (12-a) below removes accusative readings if there is an intransitive verb preceding.

(12) a. `REMOVE Acc IF (-1C Fin + IV) (NOT 0 AccAdv) ;`

b. `REMOVE TV - Pass IF (NOT 1* Acc) (NOT -1* Acc) ;`

c. *Ona smotrit vezde, no ne možet ego najti.*
She looks around, but she cannot **find him**.

When the rules in (12-b) and (12-a) are applied to the sentence in (12-c), the interaction of these rules leads to the removal of correct readings. In this sentence, *možet* 'can' is tagged as intransitive, so rule (12-a) fires, removing the accusative reading of *ego* 'him'. Once the accusative reading is removed, rule (12-b) erroneously removes the transitive reading from *najti* 'find'.

Difficult linguistics Russian participles inflect for number and case, and like all modifiers, they must agree with the noun they modify. However, participles can

[10]The rule could also be made slightly more safe with regard to unkown words by capturing the generalization that verbs with the prefix *protiv-* 'anti-, against-' tend to govern dative. Some research is needed to determine the reliability of that generalization.

also be used as substantives, so dealing with participles can be difficult in cases where the participle is in the same case as the case that it governs. Example (13-a) shows a rule that removes intransitive readings when the token is followed by an accusative.

However, example (13-b) shows a case in which this rule fails. In this example, *Vanju i Mašu* 'Vanja.ACC and Masha.ACC' are the object of *vidit* 'sees' and the participle *igrajuščix* 'playing.PL-ACC' is modifying *Vanju i Mašu*. However, the syntactic structure of the sentence technically allows for another the possibility that is only ruled out by real-world semantic knowledge. In this alternate interpretation, *igrajuščix* 'those who play' is the object of *vidit* 'sees', whereas *Vanju i Mašu* 'Vanja and Masha' is the direct object of *igrajuščix*: 'Mama sees the people who are playing Vanja and Masha.'

(13) a. `REMOVE IV IF (0 TV OR IV) (1C Acc) (NOT 1 AccAdv) ;`

b. *Ix mama vnutri doma s koškoj, ona smotrit v okno i vidit* ***igrajuščix*** *Vanju i Mašu.*
'Their mother is inside the house with the cat, she looks through the window and sees Vanja.SG-ACC and Masha.SG-ACC **playing**.PL-ACC.'

This kind of error would ideally be resolved with semantic knowledge, but in the interest of maintaining high recall, the rule can be modified to exclude contexts that exhibit this kind of clear dependency ambiguity.

3.6.3 Task-based evaluation

The constraint grammar described in this chapter has been applied to the task of automatic word stress placement, as described more fully in Chapter 4 and Reynolds and Tyers (2015). This task is especially relevant for Russian language learners, because vowels are pronounced differently depending on their position relative to stress position. For example, the word *molokó* 'milk' is pronounced /məlʌkɔ/, where each instance of the letter *o* corresponds to a different vowel sound. Russian also has complicated patterns of shifting stress, which are difficult for learners to master.

Stress can be placed in running text by sending it through our morphological analyzer and constraint grammar, and then sending the constraint grammar output back to a morphological wordform generator. Out of all the wordforms in running text for which our morphological analyzer/generator (without the CG) outputs multiple stress positions, almost 99% can be disambiguated morphosyntactically, so a constraint grammar can theoretically resolve most stress ambiguity indirectly. The

results of Reynolds and Tyers (2015) show that our constraint grammar overcomes about 42% of the ambiguity relevant to stress ambiguity in unrestricted text. A more detailed explanation of the experiment can be found in Chapter 4.

3.6.4 Combining with a statistical tagger

Given that just over half of all ambiguity remains after running our preliminary constraint grammar and that for many applications unambiguous output is necessary, we decided to experiment with combining the constraint grammar with a statistical tagger to resolve remaining ambiguity. Similar approaches have been taken by previous researchers with Basque (Ezeiza et al., 1998), Czech (Hajič et al., 2001, 2007), Norwegian (Johannessen et al., 2011, 2012), Spanish (Hulden and Francom, 2012), and Turkish (Oflazer and Tür, 1996).

We follow the voting method described by Hulden and Francom (2012). We used the freely available `hunpos` part-of-speech tagger (Halácsy et al., 2007), which is an open-source clone of the well-known TnT part-of-speech tagger (Brants, 2000). We performed 10-fold cross validation using our evaluation corpus, taking 10% for testing and 90% for training, and experimented with three configurations:

- `HMM` (Hidden Markov Model): the `hunpos` part-of-speech tagger with its default options
- `HMM+Morph`: as with `HMM` but incorporating the output of our morphological analyzer (see section 3.4.1) as a full form lexicon.
- `HMM+Morph+CG`: we submitted the output from `HMM+Morph` and the constraint grammar to a voting procedure, whereby if the constraint grammar left one valid reading, we chose that, otherwise if the constraint grammar left a word with more than one reading, we chose the result from the `HMM+Morph` tagger.

As can be seen from Figure 3.3, incorporating the constraint grammar improves the performance of the HMM tagger, an improvement of nearly 5% in accuracy, similar to that reported by Hulden and Francom (2012) for the same amount of training data. In Figure 3.3, it appears that the HMM alone is much more dependent on training corpus size than the voting setup, which improves very little between a training corpus size of 5,000 and 9,000.

Our constraint grammar also has a much lower precision as a result of the ambiguity remaining in the output. Similarly, the final accuracy is below the state of the art for Russian. For instance, Sharoff et al. (2008a) report a maximum accuracy of 95.28% using the TnT tagger. Note, however, that this model was trained on a much larger corpus – over five million tokens – which is not freely available.

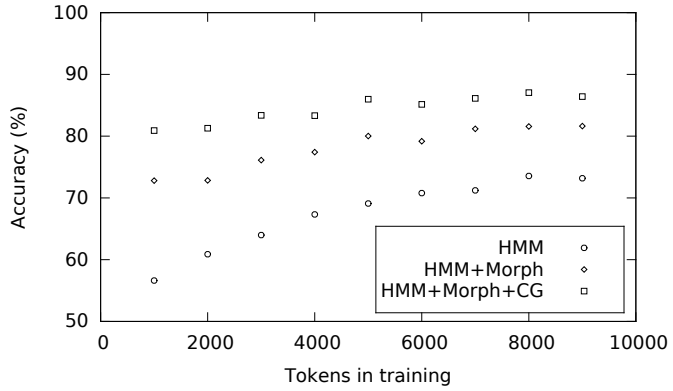


Figure 3.3: Learning curve for three tagging setups: `hunpos` with no lexicon; `hunpos` with a lexicon; and `hunpos` with a lexicon and the Russian constraint grammar in a voting set up.

3.7 Conclusions and Outlook

This chapter has presented a preliminary constraint grammar for Russian, where rules have been sorted into a reliability hierachy based on observations of performance on a non-gold-standard corpus. The constraint grammar is high recall (over 0.99) and improves the performance of a trigram HMM-based tagger. It also shows state-of-the-art performance for the stress-placement task.

It is worth noting that although Russian has a great deal of non-free resources, this chapter presents a method which is promising for smaller or lesser-resourced Slavic languages such as Sorbian, Rusyn or Belarusian. Instead of hand-annotating a large quantity of text, it may be more efficient to work on grammatical resources — such as a morphological analyzer and constraint grammar — and use them alongside a smaller quantity of high-quality annotated text.

We have a number of plans for future work, the first of which is increasing the precision of the grammar without decreasing recall. Second, we also plan to fully implement syntactic function labeling, which can then feed into dependency parsing. For the dependency parser we plan to adapt the Giellatekno dependency grammar of Antonsen et al. (2010), which describes porting a North Sami dependency grammar to four other languages.

Our development workflow could also be improved. For example, during the testing of each rule we could save the correct decisions of the grammar. This would give us a partially-disambiguated development corpus, which could be gradually used to build up a corpus for regression testing to ensure that new rules added do not invalidate the correct decisions of previously written rules. This corpus could

potentially also be used as a gold-standard corpus, although testing methodologies would need to take into account the fact that the corpus was selected using a non-random process.

Part II

Applications of the analyzer in language learning

Chapter 4

Automatic stress placement in unrestricted text[1]

We evaluate the effectiveness of finite-state tools we developed for automatically annotating word stress in Russian unrestricted text, i.e. running text. This task is relevant for computer-assisted language learning and text-to-speech. To our knowledge, this is the first study to empirically evaluate the results of this task. Given an adequate lexicon with specified stress, the primary obstacle for correct stress placement is disambiguating homographic wordforms. The baseline performance of this task is 90.07%, (known words only, no morphosyntactic disambiguation). Using a constraint grammar to disambiguate homographs, we achieve 93.21% accuracy with minimal errors. For applications with a higher threshold for errors, we achieved 96.15% accuracy by incorporating frequency-based guessing and a simple algorithm for guessing the stress position on unknown words. These results highlight the need for morphosyntactic disambiguation in the word stress placement task for Russian, and set a standard for future research on this task.

4.1 Introduction

Lexical stress and its attendant vowel reduction are a prominent feature of spoken Russian; the incorrect placement of stress can render speech almost incomprehensible. This is because Russian word stress is phonemic, i.e. many wordforms are distinguished from one another only by stress position. Half of the vowel letters in Russian change their pronunciation significantly, depending on their position

[1]The research in this chapter was carried out in collaboration with Francis Tyers. It is a modified version of Reynolds and Tyers (2015). Most of the work is my own, but Dr. Tyers wrote the first version of the script to perform basic stress annotation.

relative to the stress. For example the word *dogovórom* 'contract.SG-INS' is pronounced /dəgʌvɔrəm/, with the letter *o* realized as three different vowel sounds. Determining these vowel qualities is impossible without specifying the stress position, but standard written Russian does not typically mark word stress.[2] Without information about lexical stress position, correctly converting written Russian text to speech is impossible.

Determining stress position is a problem both for humans (e.g. foreign language learners) and computers (e.g. text-to-speech). This can be the cause of considerable difficulty for learners, since the inflecting word classes include complex patterns of shifting stress, and a lexeme's stress pattern cannot be predicted from surface forms.

Stress position ambiguity corresponds to multiple kinds of lexical and morphosyntactic ambiguity. We identify three different types of relations between word stress ambiguity and morphosyntactic ambiguity. First, *intraparadigmatic* stress ambiguity refers to homographic wordforms belonging to the same lexeme, as shown in (1).[3]

(1) Intraparadigmatic homographs
   a. *téla* 'body.$_{\text{SG-GEN}}$'
   b. *telá* 'body.$_{\text{PL-NOM}}$'

The remaining two types of stress ambiguity occur between lexemes. *Morphosyntactically incongruent* lexical ambiguity occurs between homographs that belong to separate lexemes, and whose morphosyntactic values are different, as shown in (2).

(2) Morphosyntactically incongruent homographs
   a. *nášej* 'our.$_{\text{F-SG-GEN/LOC/DAT/INS}}$'
      našéj 'sew on.$_{\text{IMP-2SG}}$'
   b. *dorógа* 'road.$_{\text{N-F-SG-NOM}}$'
      dorogá 'dear.$_{\text{ADJ-F-SG-PRED}}$'

Morphosyntactically congruent lexical ambiguity occurs between homographs that belong to separate lexemes, and whose morphosyntactic values are identical, as shown in (3). This kind of ambiguity is relatively rare, and resolving this ambiguity is best achieved by means of technologies such as word sense disambiguation.

[2] Texts intended for native speakers sometimes mark stress on words that cannot be disambiguated through context. Theoretically, a perfect word stress placement system could help an author identify tokens which should be stressed for natives: any token that cannot be disambiguated by syntactic or semantic means should be marked for stress.

[3] Throughout this chapter, cyrillic is transliterated using the scientific transliteration scheme.

(3) Morphosyntactically congruent homographs
   a. *zámok* 'castle.$_{\text{SG-NOM}}$'
      zamók 'lock.$_{\text{SG-NOM}}$'
   b. *zámkov* 'castle.$_{\text{PL-GEN}}$'
      zamkóv 'lock.$_{\text{PL-GEN}}$'

It should be noted that most morphosyntactic ambiguity in unrestricted text does not result in stress ambiguity. For example, *novyj* 'new' (and every other adjective) has identical forms for F-SG-GEN, F-SG-LOC, F-SG-DAT and F-SG-INS: *nóvoj*. Likewise, the form *vypej* has multiple possible readings (including 'drink.$_{\text{IMP}}$', 'bittern.$_{\text{PL-GEN}}$'), but they all have the same stress position: *výpej*. We refer to this as *stress-irrelevant* morphosyntactic ambiguity, since all readings have the same stress placement.

In the case of unrestricted text in Russian, most stress placement ambiguity is rooted in intraparadigmatic and morphosyntactically incongruent lexical ambiguity. Detailed part-of-speech tagging with morphosyntactic analysis can help determine the stress of these forms, since each alternative stress placement is tied to a different tag sequence. In this study we focus on the role of detailed part-of-speech tagging in improving automatic stress placement. We leave morphosyntactically congruent stress ambiguity to future work because it is by far the least common type of stress ambiguity (less than 1% of tokens in running text), and disambiguating morphosyntactically congruent stress requires fundamentally different technology from the other approaches of this study.

4.1.1 Background and task definition

Automatic stress placement in Russian is similar to diacritic restoration, a task which has received increasing interest over the last 20 years. Missing diacritics can complicate many NLP tasks, such as text-to-speech, since generally speaking, diacritics disambiguate otherwise homographic wordforms. For example, speakers of Czech may type emails and other communications without standard diacritics. In order to generate speech from these texts, they must first be normalized by restoring diacritics.

A slightly different situation arises with languages whose standard orthography is underspecified, like vowel quality in Arabic or Hebrew. For such languages, the "restoration" of vowel diacritics results in text that is overspecified with regard to standard orthography. For languages with inherently ambiguous orthography, it may be more precise to refer to this as "diacritic enhancement', since it produces text that is less ambiguous than the standard language. In this sense, Russian orthography is similar to Arabic and Hebrew, since its vowel qualities are also

underspecified in standard orthography.

Many studies of Russian text-to-speech and automatic speech recognition make note of the difficulties caused by the shortcomings of their stress-marking resources (e.g. Krivnova, 1998). Text-to-speech technology must deal with the inherent ambiguity of Russian stress placement, and many articles mention disambiguation of one kind or another, but to our knowledge no studies have empirically evaluated the success of their approaches.

Several studies have investigated methods for predicting stress position on unknown words. For example, Xomicevič et al. (2008) developed a set of heuristics for guessing stress placement on unknown words in Russian. More recently, Hall and Sproat (2013) trained a maximum entropy model on a dictionary of Russian words, and evaluated on wordlists containing 'known' and 'unknown' wordforms.[4] Their model achieved 98.7% accuracy on known words, and 83.9% accuracy on unknown words. Note that the task of training and evaluating on wordlists is different from that of placing stress in running text. Since many of the most problematic stress ambiguities in Russian occur in high-frequency wordforms, evaluations of wordform lists encounter stress ambiguity seven times less frequently than in running text (see discussion in Section 4.4). Hall and Sproat (2013) do not make use of morphosyntactic information in their model.

So far, the implicit target application of the few studies related to automatic stress placement in Russian has been text-to-speech and automatic speech recognition. However, the target application of our stress annotator is in a different domain: language learning. Since standard Russian does not mark word stress, learners are frequently unable to pronounce unknown words correctly without referencing a dictionary or similar resources. In the context of language learning, marking stress incorrectly is arguably worse than not marking it at all. Because of this, we designed our stress annotator to be adjustable according to various confidence thresholds. For language-learning applications, we want the annotator to abstain from marking stress on words that it is unable to resolve with high confidence.

4.1.2 Stress corpus

Russian texts with marked word stress are relatively rare, except in materials for second language learners, which are predominantly proprietary. Our gold-standard corpus was collected from free texts on Russian language-learning websites. This

[4] Hall and Sproat (2013) randomly selected their training and test data from a list of wordforms, and so a number of lexemes had wordforms in both the training and test data. Wordforms in the test data whose sibling wordforms from the same lexeme were in the training set were categorized as 'known' wordforms.

small corpus (7689 tokens) is representative of texts that learners of Russian are likely to encounter in their studies. These texts include excerpts from well-known literary works, as well as dialogs, prose, and individual sentences that were written for learners.

Unfortunately, the general practice for marking stress in Russian is to *not* mark stress on monosyllabic tokens, effectively assuming that all monosyllabics are stressed. However, this approach is not well-motivated. Many words—both monosyllabic and multisyllabic—are unstressed, especially among prepositions, conjunctions, and particles. Furthermore, there are many high-frequency monosyllabic homographs that can be either stressed or unstressed, depending on their part of speech, or their appearance in particular collocations. For example, the token *čto* is stressed when it means 'what' and unstressed in the conjunction *potomu čto* 'because'. For such words, one cannot simply assume that they are stressed on the basis of their syllable count.

Based on these considerations, we built our tools to mark stress on every word, both monosyllabic and multisyllabic. However, because our gold-standard corpus texts do not mark stress on monosyllabic words, we cannot evaluate our annotation of those words.

Similarly, some compound Russian words have secondary stress, but this is rarely marked, if at all, even in educational materials. Therefore, even though our tools are built to mark secondary stress, we cannot evaluate secondary stress marks, since they are absent in our gold-standard corpus.

In order to test our word stress placement system, we removed all stress marks from the gold-standard corpus, then marked stress on the unstressed version using our tools, and then compared with the stress-marked version in the gold standard.

4.2 Automatic stress placement

State-of-the-art morphological analysis in Russian is based on finite-state technology (Nozhov, 2003; Segalovich, 2003). As described in Chapter 2, I developed free and open-source finite-state tools capable of analyzing and generating stressed wordforms. This morphological analyzer is described in detail in Chapter 2, but in short, it is a Finite-State Transducer[5] (FST), and based on the well-known *Grammatical Dictionary of Russian* (Zaliznjak, 1977). The FST outputs all possible morphosyntactic readings of each wordform, and our Constraint Grammar[6] (Karlsson, 1990; Karlsson et al., 1995) then removes some readings based on syntactic

[5]Using two-level morphology (Koskenniemi, 1983, 1984), implemented in both xfst (Beesley and Karttunen, 2003) and hfst (Linden et al., 2011)

[6]Implemented using vislcg3 constraint grammar parser (http://beta.visl.sdu.dk/cg3.html).

context. The constraint grammar is described in more detail in Chapter 3.

The ultimate success of our stress placement system depends on the performance of the constraint grammar. Ideally, the constraint grammar would successfully remove all but the correct reading for each token, but in practice some tokens still have more than one reading remaining. Therefore, we also evaluate various approaches to deal with the remaining ambiguity, as described below. Table 4.1 shows two possible sets of readings for the token *kosti*, as well as the output of each approach described below. The token *kosti* can belong to the lexeme *kost'* 'bone', or the verb *kostit'*, a less common word meaning 'chew out'. The first column exhibits stress ambiguity between the noun readings and the imperative verb reading. The second column shows a similar set of readings, after the constraint grammar has removed the imperative verb reading. This results in only stress-irrelevant ambiguity.

Readings:	кость-N-F-SG-GEN *kósti* кость-N-F-SG-DAT *kósti* костить-V-IPFV-IMP *kostí*	кость-N-F-SG-GEN *kósti* кость-N-F-SG-DAT *kósti*
`bare`	*kosti*	*kosti*
`safe`	*kosti*	*kósti*
`randReading`	*kósti* (p=0.67) or *kostí* (p=0.33)	*kósti*
`freqReading`	*kósti*	*kósti*

Table 4.1: Example output of each stress placement approach, given a particular set of readings for the token *kosti*

The `bare` approach is to not mark stress on words with more than one reading. Since both sets of readings in Table 4.1 have more than one reading, `bare` does not output a stressed form.

The `safe` approach is to mark stress only on tokens whose morphosyntactic ambiguity is stress-irrelevant. In Table 4.1, the first column has readings that result in two different stress positions, so `safe` does not output a stressed form. However, in the second column, both readings have the same stress position, so `safe` outputs that stress position.

The `randReading` approach is to randomly select one of the available readings. In the first column of Table 4.1, a random selection means that *kósti* is twice as likely as *kostí*, since two readings give *kósti* and one reading yields *kostí*. The second column of Table 4.1 contains stress-irrelevant ambiguity, so a random selection of a reading has the same result as the `safe` approach.

The `freqReading` approach is to select the reading that is most frequent, with frequency data taken from a separate hand-disambiguated corpus. If none of the readings are found in the corpus, then `freqReading` selects the reading with the

tag sequence (lemma removed) that is most frequent in our corpus. If the tag sequence is not found in our frequency list, then `freqReading` backs off to the `randReading` algorithm. In the first column of Table 4.1, `freqReading` selects *kósti* because the tag sequence N-F-SG-GEN is more frequent than the other alternatives. Note that for tokens with stress-irrelevant ambiguity (e.g. the second column of Table 4.1), `randReading` and `freqReading` produce the same result as the `safe` method.

So far, the approaches discussed are dependent on the availability of readings from the FST. The focus of our study is on disambiguation of known words, but we also wanted to guess the stress of unknown tokens in order to establish some kind of accuracy maximum for applications that are more tolerant of higher error rates. To this end, we selected a simple guessing method for unknown words. A recent study by Lavitskaya and Kabak (2014) concludes that Russian has default final stress in consonant-final words, and penultimate stress in vowel-final words.[7] Based on this conclusion, the `guessSyll` method places the stress on the last vowel that is followed by a consonant.[8] This method is applied to unknown wordforms in two approaches, `randReading+guessSyll` and `freqReading+guessSyll`, which are otherwise identical to `randReading` and `freqReading`, respectively. For known tokens, these methods select either a random reading or the most frequent (respectively), but for unknown tokens, they use the `guessSyll` approach.

For our baseline, we take the output of our morphological analyzer (without the constraint grammar) in combination with the `bare`, `safe`, `randReading`, `freqReading`, `randReading+guessSyll`, and `freqReading+guessSyll` approaches. We also compare our outcomes with the RussianGram[9] plugin for the Google Chrome web browser. RussianGram is not open-source, so we can only guess what technologies support the service. In any case, it provides a meaningful reference point for the success of each of the methods described above.

4.3 Results

We evaluated all multisyllabic words with marked stress in the gold-standard corpus (N = 4048). Since our approach is lexicon-based, some of our results should be interpreted with respect to how many of the stressed wordforms in the gold-standard corpus can be found in the output of the morphological analyzer. We refer

[7] There is some disagreement over how to define default stress in Russian, cf. Crosswhite et al. (2003).

[8] Although this approach is simplistic, unknown words are not the central focus of this study. More sophisticated heuristics and machine-learning approaches to unknown words are discussed in Section 4.4.

[9] http://russiangram.com/

to this measure as *recall*.[10] Out of 4048 tokens, 3949 were found in the FST, which is equal to 97.55%. This number represents the upper bound for methods relying on the FST. Higher scores are only achievable by expanding the FST's lexicon or by using syllable-guessing algorithms. After running the constraint grammar, recall was 97.35%, a reduction of 0.20%. This is presumably the result of the grammar removing a small number of valid readings.

Results were compiled for each of the 13 approaches discussed above: *without* the constraint grammar (`noCG`) x 6 approaches, *with* the constraint grammar (`CG`) x 6 approaches, and RussianGram (`RussianGram`). Results are given in Table 4.2. Each token was categorized as either an accurate output, or one of two categories of failures: errors and abstentions. If the stress tool outputs a stressed wordform, and it is incorrect, then it is counted as an 'error' (err). If the stress tool outputs an unstressed wordform, then it is counted as an 'abstention' (abs). Abstentions can be the result of either unknown wordforms, or known wordforms with no stress specified in our lexicon.

approach	acc%	err%	abs%	totTry%	totFail%
`noCG+bare`	30.43	0.17	69.39	30.61	69.57
`noCG+safe`	90.07	0.49	9.44	90.56	9.93
`noCG+randReading`	94.34	3.36	2.30	97.70	5.66
`noCG+freqReading`	95.53	2.59	1.88	98.12	4.47
`noCG+randReading+guessSyll`	94.99	4.05	0.96	99.04	5.01
`noCG+freqReading+guessSyll`	95.83	3.46	0.72	99.28	4.17
`CG+bare`	45.78	0.44	53.78	46.22	54.22
`CG+safe`	93.21	0.74	6.05	93.95	6.79
`CG+randReading`	95.50	2.59	1.90	98.10	4.50
`CG+freqReading`	95.73	2.40	1.88	98.12	4.27
`CG+randReading+guessSyll`	95.92	3.33	0.74	99.26	4.08
`CG+freqReading+guessSyll`	96.15	3.14	0.72	99.28	3.85
`RussianGram`	90.09	0.79	9.12	90.88	9.91

Table 4.2: Results of stress placement task evaluation

The two right-most columns in Table 4.2 combine values of the basic categories. The term 'totTry' ('total attempted') refers to the sum of the accuracy and

[10] Our method of computing recall assumes that if even one reading is output by the FST, then all possible readings are present. If the token *kosti* were in our corpus, and its correct reading were the verbal imperative, as shown in Table 4.1, and if the FST hypothetically did not contain this reading, then our method for computing recall would assume that the token was found in the FST, even though a correct alternative exists. We have not attempted to formally estimate how frequently this assumption fails, but we expect such cases to be rare.

error rate. This number represents the proportion of tokens for which our system outputs a stressed wordform. In the case of `noCG+bare`, the acc% (30.43) and err% (0.17) sum to the totTry% value of 30.61. The term 'totFail' ('total failures') refers to the sum of error rate and abstention rate, which is the proportion of tokens for which the system failed to output the correct stressed form. In the case of `noCG+bare`, the err% (0.17) and abs% (69.39) sum to the totFail% value of of 69.57 (rounded).

The `noCG+bare` approach achieves a baseline accuracy of 30.43%, so roughly two thirds of the tokens in our corpus are morphosyntactically ambiguous. The error rate of 0.17% primarily represents forms whose stress position varies from speaker to speaker (e.g. *zavílis'* vs. *zavilís'* 'they crinkled'), or errors in the gold-standard corpus (e.g. **verím* 'we believe', which should be *vérim*).

The `noCG+safe` approach achieves a 60% improvement in accuracy (90.07%), which means that 89.39% of morphosyntactic ambiguity on our corpus is stress-irrelevant. Notably, the RussianGram web service achieves results that are very similar to the `noCG+safe` approach.

Since the ceiling recall for the FST is 97.55%, and since the `noCG+safe` approach achieves 90.07%, the maximum improvement that a constraint grammar could theoretically achieve is 7.48%. A comparison of `noCG+safe` and `CG+safe` reveals an improvement of 3.14%, which is about 42% of the way to the ceiling recall.

The `CG+randReading` and `CG+freqReading` approaches are also limited by the 97.55% ceiling from the FST, and their accuracies achieve improvements of 2.29% and 2.52%, respectively, over `CG+safe`. However, these gains come at the cost of error rates as much as 3.5 times higher than `CG+safe`: +1.85% and +1.66%, respectively. It is not surprising that `CG+freqReading` has higher accuracy and a lower error rate than `CG+randReading`, since frequency-based guesses are by definition more likely to be correct than random guesses. The frequency data were taken from a very small corpus, and it is likely that frequency from a larger corpus would yield even better results.

The `guessSyll` approach was designed to make a guess on every wordform that is not found in the FST, which would ideally result in an abstention rate of 0%. However, the abstention rates of approximately 0.7% are a manifestation of the fact that some words in the FST, especially proper nouns, have not been assigned stress. Because the FST outputs a form – albeit unstressed – the `guessSyll` algorithm is not called. This means that `guessSyll` is only guessing on about 2% of the tokens. The improvement on overall accuracy from `CG+freqReading` to `CG+freqReading+guessSyll` is 0.42%, which means that the `guessSyll` method guess was accurate 21% of the time.

4.4 Discussion

One of the main points of this chapter is to highlight the importance of syntactic context in the Russian word stress placement task. If your intended application has a low tolerance for error, the `noCG+safe` approach represents the highest accuracy that is possible without leveraging syntactic information for disambiguation (90.07%). In other words, a system that is blind to morphosyntax and contextual disambiguation cannot significantly outperform `noCG+safe`. It would appear that this is the method used by `RussianGram`, since its results are so similar to `noCG+safe`. Indeed, this result can be achieved most efficiently without any part-of-speech tagging, through simple dictionary lookup.

We noted in Section 4.1.1 that Hall and Sproat (2013) achieved 98.7% accuracy on stress placement for individual wordforms in a list (i.e. *without* syntax). This result is 8.63% higher than `noCG+safe`, but it is also a fundamentally different task. Based on the surface forms in our FST – which is based on the same dictionary used for Hall and Sproat (2013) – we calculate that only 29 518 (1.05%) of the 2 804 492 wordforms contained in our FST are stress-ambiguous.[11] In our corpus of unrestricted text, at least 7.5% of the tokens are stress-ambiguous. Therefore, stress ambiguity is more than seven times more prevalent in our corpus of unrestricted text than it is in our wordform dictionary. Since the task of word stress placement is virtually always performed on running text, it seems prudent to make use of surrounding contextual information. The experiment described in this chapter demonstrates that a constraint grammar can effectively improve the accuracy of a stress placement system without significantly raising the error rate. Our Russian constraint grammar is under continual development, so we expect higher accuracy in the future.

We are unaware of any other empirical evaluations of Russian word stress placement in unrestricted text. The results of our experiment are promising, but many questions remain unanswered. The experiment was limited by properties of the gold-standard corpus, including its size, genre distribution, and quality. Our gold-standard corpus represents a broad variety of text genres, which makes our results more generalizable, but a larger corpus would allow for evaluating each genre individually. For example, the vast majority of Russian words with shifting stress are of Slavic origin, so we expect a genre such as technical writing to have a lower proportion of words with stress ambiguity, since it contains a higher proportion of borrowed words, calques, and neologisms with simple stress patterns.

In addition to genre, it is also likely that text complexity affects the difficulty of

[11] These stress-ambiguous wordforms have an average of 2.01 possible stress positions, so a perfect stress placement tool has a 50/50 chance of getting them right. This means that a perfect stress placement tool could achieve 99.48% on their task.

the stress placement task. The distribution of different kinds of syntactic constructions varies with text complexity (Vajjala and Meurers, 2012), and so we expect that the effectiveness of the constraint grammar will be affected by those differences.

The resources needed for machine-learning approaches to this task – such as a large corpus of Russian unrestricted text with marked stress – are simply not available at this time. Even so, lexicon- and rule-based approaches have some advantages over machine-learning approaches. For example, we are able to abstain from marking stress on tokens whose morphosyntactic ambiguity cannot be adequately resolved by linguistically motivated rules. In language-learning applications, this reduces the likelihood of learners being exposed to incorrect wordforms, and accepting them as authoritative. Such circumstances can lead to considerable frustration and lack of trust in the learning tool. However, in error-tolerant applications, machine-learning does seem well-suited to placing stress on unknown words, since morphosyntactic analysis is problematic.

The syllable-guessing algorithm `guessSyll` used in this experiment was overly simplistic, and so it was not surprising that it was only moderately successful. More rigorous rule-based approaches have been suggested in other studies (Church, 1985; Williams, 1987; Xomicevič et al., 2008). For example, Xomicevič et al. (2008) attempt to parse the unknown token by matching known prefixes and suffixes.

Other studies have applied machine-learning to the task of guessing stress of unknown words (Pearson et al., 2000; Webster, 2004; Dou et al., 2009; Hall and Sproat, 2013). For example, Hall and Sproat (2013) achieve an accuracy of 83.9% with unknown words. Unfortunately, their model was trained on a full list of Russian words, which is not representative of the words that would be unknown to a system like ours. However, their approach could be modified to make a guesser that is tuned to the 'clean-up' task in our application. Most of the complicated word stress patterns are closed classes which we expect our analyzer to cover with 100% accuracy.[12] By excluding closed classes of words from the training data, and including word classes that are likely to be similar to unknown tokens, such as those with productive derivational affixes, we might be able to outperform a model that is trained on a full wordlist.

4.5 Conclusions

We have demonstrated the effectiveness of using a constraint grammar to improve the results of a Russian word stress placement task in unrestricted text by resolving

[12]The growing number of masculine nouns with shifting stress (*dóktor∼doktorá* 'doctor'∼'doctors') is one exception to this generalization.

42% of the stress ambiguity in our gold-standard corpus. We showed that stress ambiguity is seven times more prevalent in our corpus of running text than it is in our lexicon, suggesting the importance of context-based disambiguation for this task. As with any lexicon- and rule-based system, the lexicon and rules can be expanded and improved, but our initial results are promising, especially considering the short timespan over which our tools were developed.

As this is the first empirical study of its kind, we also discussed some methodological limitations, as well as possible improvements for subsequent research. These include collecting stressed corpora of varying text complexity and/or genre, as well as implementing and/or adapting established word stress-guessing methods for unknown words.

The motivation for developing technology to automatically annotate Russian word stress in this study was to make it possible for language learners to have access to information about word stress in texts that have not been hand-annotated by a teacher or publisher. The following chapter introduces an application that was developed to facilitate this and other functionalities for teachers and learners.

Chapter 5

Visual Input Enhancement of the Web[1]

In this chapter, we explore the challenges and opportunities which arise in developing automatic visual input enhancement activities for Russian with a focus on target selection and adaptive feedback. Russian, a language with a rich fusional morphology, has many syntactically relevant forms that are not transparent to the language learner. This makes it a good candidate for visual input enhancement (VIE). VIE essentially supports incidental focus on form by increasing the salience of language forms to support noticing by the learner. The freely available VIEW system (Meurers et al., 2010) was designed to automatically generate VIE activities from any web content. We extend VIEW to include several Russian grammar topics and discuss connected research issues regarding target selection, ambiguity management, prompt generation, and distractor generation. We show that the same information and techniques used for target selection can often be repurposed for adaptive feedback. Authentic Text ICALL (ATICALL) systems incorporating only native-language processing technology—without the learner-language processing technology that is characteristic of Intelligent Language Tutoring Systems (ILTS)—thus can support some forms of adaptive feedback. This demonstrates that ATICALL and ILTS represent a spectrum of possibilities rather than two categorically distinct enterprises.

[1]The research presented in this chapter was carried out in collaboration with Eduard Schaf, java programmer and undergraduate student, as well as Prof. Detmar Meurers, who helped to edit, revise, and even coauthor some passages in the article on which this chapter is based (Reynolds et al., 2014).

5.1 Introduction

Intelligent Computer-Assisted Language Learning (ICALL) has been characterized (Meurers, 2012) as consisting of two distinct areas, Intelligent Language Tutoring Systems (ILTS) and Authentic Text ICALL (ATICALL). In the former, researchers have focused on the challenge of analyzing learner language and providing adaptive feedback. Analyzing learner language requires language models that incorporate expected learner errors of various kinds, including lexical, morphological and syntactic. Learner-language models can be used in ICALL settings, as well as in automated scoring, and the analysis and annotation of learner corpora. On the other hand, ATICALL research employs standard Natural Language Processing (NLP) tools developed for analyzing native language to identify and enhance authentic texts in the target language. The use of native-language NLP tools facilitates the search for and enhancement of reading material, identifying relevant examples of grammatical structures in native-language corpora, and generating exercises, games, and tests from authentic materials.

While the choice of NLP may seem to result in categorically different functionality in some respects, in this chapter we want to show that it is possible to achieve very similar functionality with both approaches. We describe how an ATICALL system can incorporate a feature typical of ILTS: adaptive feedback to learner responses. The same technology used to generate grammar exercises can be extended to provide adaptive feedback to learners. The idea is explored using language activities for Russian, a language with a rich, fusional morphology that is challenging for second language learners. We showcase four of the Russian activities that we developed on top of the freely available VIEW platform (Meurers et al., 2010).

Russian Morphological Analysis Most Russian grammar books focus primarily on morphology, a serious challenge to most learners. Russian has a highly fusional morphology, with nominal inflection for six cases, two numbers, and three genders. There are three noun declension paradigms (i.e., inflection classes), each containing 12 forms. Adjectival modifiers have at least 24 forms. Russian verbs represent a relatively extensive inflectional system, similar to other Indo-European languages. Including participial declension, verbs can have more than a hundred paradigm cells. A related difficulty is that Russian stress is phonemic, differentiating both lexical and inflectional homographs. This causes difficulties for learners, since there is a complex system of lexically specified stress placement, yet stress is almost never marked in the written language.

In order to build an ATICALL system for Russian, we needed a fast, broad-coverage morphological engine to both analyze and generate word forms (with marked lexical stress). A Russian finite-state morphological analyzer was developed—

as described in Chapter 2—using the two-level formalism (Koskenniemi, 1983). The transducer was originally based on Zaliznjak (1977) ($\approx$ 120 000 words), which is the foundation for most Russian computational morphologies. Additional words, especially proper nouns, are continually being added. Since Russian has systematic syncretism and widespread homonymy, a constraint grammar (Karlsson et al., 1995) implemented in the freely available CG3 system[2] is used to disambiguate multiple readings. The grammar is also under continual development. A description and evaluation of the grammar is given in Chapter 3. Our constraint grammar is tuned to achieve high recall, which means that it should only remove readings that can be ruled out with very high confidence. The benefit of this bias is that downstream processes can operate with an intuitionistic/epistemic logic, effectively allowing the system to "know whether it knows" that a given reading is reliable.

Most state-of-the-art part of speech taggers for Russian are based on finite-state transducers, including *AOT/Dialing* (Nozhov, 2003), and *mystem* (Segalovich, 2003). The benefits of a rule-based approach to Russian morphology is even more pronounced in ATICALL applications. Finite-state methods make it possible to provide efficient and robust computational analyses with wide empirical coverage, while keeping a clear conceptual distinction between the linguistic system and its usage. Finite-state methods also have several characteristics that make them especially well suited for ICALL. A constraint grammar-based analysis can "know whether it knows", which allows an ATICALL system to focus only on targets that are clearly identifiable. Since finite-state tools provide an actual linguistic model of the language being analyzed, it is possible to identify and increase the salience of linguistic characteristics known to be relevant in language learning. A good case in point is stress placement, which is lexical, yet requires syntactic disambiguation. The lexicon of the finite-state analyzer provides effective access to subsets of data for certain grammar topics (e.g., retrieve all words with a particular stress pattern), since this information is modeled in the analyzer's source files. Furthermore, mistakes and errors in the system can be diagnosed and corrected. This is especially important in ICALL, where low precision in the analysis leads to unreliable output easily confusing and frustrating learners. And, importantly in the context of ATICALL involving activities with distractors, a finite-state morphological analyzer can simply be reversed to become a generator.

Visual Input Enhancement Researchers in second language acquisition agree that comprehensible input is necessary for language learning. The Noticing Hypothesis (Schmidt, 1990) extends this claim to say that noticing of grammatical

[2] `http://beta.visl.sdu.dk/cg3.html`

categories and relations also is required for successful second language acquisition. Based on the Noticing Hypothesis and related work on Consciousness Raising (Rutherford and Sharwood Smith, 1985) and Input Enhancement (Sharwood Smith, 1993, p. 176), researchers have investigated Visual Input Enhancement (VIE) to encourage learners to notice the grammatical forms in comprehensible input. VIE refers to the graphical enhancement of written text to draw attention to targeted grammatical structures. Various modes of enhancement have been suggested, such as font manipulation (e.g., bold, italic, color), capitalization, and other notations (e.g., underlining, circling). Such textual enhancements are intended to increase the likelihood that the learner will notice the target grammatical form in its grammatical and functional context of use.

Visual Input Enhancement of the Web (VIEW) is an ATICALL system designed to automatically generate learning activities from user-selected texts on the web. A description of the system architecture can be found in Meurers et al. (2010). VIEW includes four activity types to guide the learner from recognition via practice to production. The *highlight* activity adds color to target wordforms. The *click* activity allows the learner to identify target wordforms in the text. The *multiple-choice* activity provides controlled practice, allowing the learner to choose the correct form from a multiple-choice list. The *practice* activity asks learners to type the wordforms themselves. The activities can be accessed as a web application on a webpage or through a toolbar provided as a Firefox web browser Add-on. Activities have previously been developed for English, German, and Spanish. The open-source research prototype is available at `http://purl.org/icall/view`.

The following issues were considered in developing the activities for Russian:

1. Learner needs: What are the needs of the learner?
2. Technological feasibility: Can the target construction be reliably identified using NLP?
3. Target selection: Which tokens of the target construction should be focused on?
4. Prompt generation: What kind of prompt can sufficiently constrain the learner productions for practice? (cf. Amaral and Meurers, 2011, sec. 3.1)
5. Generation of distractors for multiple-choice activities: What forms can or should serve as distractors? How does Second Language Acquisition (SLA) research help us with this, and how does the systematicity of the linguistic system allow us to generate distractors?
6. Feedback: What kind of feedback does the learner receive for (in)correct answers, under a perspective conceiving of feedback as *scaffolding* guiding the learner in their Zone of Proximal Development (Vygotsky, 1986)?

Related work To our knowledge, only two other Russian ICALL projects have been reported in the literature. One set of studies has been dedicated to a Russian intelligent language tutoring system, the Boltun project[3]. One of the Boltun project's goals is to develop learner-language NLP resources (Dickinson and Herring, 2008a,b; Dickinson, 2010). In this respect, it is not directly relevant to the research presented in this chapter.

Another project, KLIOS, was a learning management system developed specifically for Russian foreign language learning (Gorisev et al., 2013). KLIOS apparently makes use of the existing general-purpose tagger pymorphy2[4] and parser ABBYY Compreno[5], but it does not appear to incorporate any ATICALL elements. The native-language NLP tools are used to analyze learner language in responses to hand-written exercises. Unfortunately, the KLIOS project has been suspended.

Goal and Structure of the Paper The goal of this chapter is to explore the ability of authentic text ICALL systems to provide adaptive feedback to learners. In doing so, we also demonstrate some features of the Russian VIEW system that we are currently developing, for which a prototype can be found at `http://purl.org/icall/rusVIEW`. In Section 5.2, we introduce exercises for four separate target grammatical topics: Stress, Noun Declension, Aspect, and Participles. For each topic, we discuss the pedagogical motivation for the exercises, as well as relevant practical and theoretical issues that arose during development. Special attention is given to factors involved in target selection since these factors become relevant in the subsequent discussion. In Section 5.3, we show how the same technology and strategies used in target selection can be used to provide adaptive feedback. Section 5.4 summarizes the contributions of the chapter and considers options for evaluating the approach.

5.2 Key topics for Russian learners

The following grammar topics are generally difficult for learners, relatively ubiquitous in Russian text, and they allow us to exemplify central issues in visual input enhancement and the computational modeling it is built on. Section 5.2.1 introduces a basic example, highlighting the morphological analysis in a noun declension activity. The discussion of target selection for this activity illustrates the need to distinguish between grammatically and referentially determined morphosyntactic properties. Section 5.2.2 discusses activities for word stress, where target selec-

[3] `http://cl.indiana.edu/~boltundevelopment`
[4] `https://pymorphy2.readthedocs.org`
[5] `http://www.abbyy.ru/isearch/compreno`

tion is primarily lexical, but is also concerned with managing the ambiguity that arises in rule-based morphologies. Section 5.2.3 outlines verbal aspect activities, where target selection is complicated by limitations in determining whether the learner should be able to deduce the aspect of each token. Section 5.2.4 presents activities for participles, which demonstrate a more complicated use of wordform generation for providing prompts to guide learners' responses in multiple-choice and cloze activities.

In contrast to Intelligent Tutoring Systems, ATICALL systems such as VIEW and reading support tools such as Glosser-RuG (Nerbonne et al., 1998), COMPASS (Breidt and Feldweg, 1997), REAP[6], or ALPHEIOS[7] focus on the analysis of authentic native text. Where input enhancement and reading support turns into exercise generation, such as the multiple-choice and cloze activities of VIEW, the feedback currently provided by the system is very limited. If a response is correct, then it turns green. If a response is incorrect, it turns red. VIEW does not attempt to reveal *why* a response is correct or incorrect. One goal of this study was to determine to what extent we can provide more informative adaptive feedback.

In the following subsections, we consider the degree to which the feedback that learners receive in an ATICALL environment can be enhanced without developing new NLP tools for learner language analysis. For the feedback methods discussed below, enriched feedback can be provided using only the information already used in the target selection and distractor generation processes. In other words, the information used to select a given token is generally the same information that is needed to provide enriched feedback beyond a simple correct/incorrect indicator.

5.2.1 Noun declension

The relatively extensive nominal inflection system is one of the first major hurdles for most Russian learners. Learners whose L1 does not have similar noun declension frequently seem to ignore inflectional endings. A visual input enhancement activity has the potential to boost learning by raising awareness of those endings.

We developed numerous activities targeting specific case distinctions known to be difficult, including a highlight activity that highlights all of the nouns in a given case; a click activity, where learners are asked to find and click nouns in a given case; a multiple choice activity, where learners can select from all the forms of a given noun; and a practice activity, in which learners must type the noun in full. In this chapter, we focus on describing the multiple-choice activity developed for all cases, since that activity makes it possible to illustrate both the underlying NLP and some points regarding target selection. When learners select this activity for a

[6] http://reap.cs.cmu.edu
[7] http://alpheios.net

web page, VIEW replaces some nouns in the text with dropdown boxes containing the original noun in all of its case forms as options.

Target selection As a rule, each noun declension paradigm has 12 cells (six cases, singular and plural), but some forms are syncretic. For example, prototypical masculine nouns have ten unique forms, feminine and neuter nouns have nine, and the soft-consonant feminine nouns have only seven unique forms. Although our constraint grammar is able to disambiguate many syncretic forms, some ambiguity still remains in the output for many tokens. One might expect that ambiguity in the analysis would complicate target selection, but this is only true if the analysis is ambiguous with regard to number. This is because a number ambiguity may be a referential ambiguity that usually cannot be resolved by checking contextual clues, as illustrated in (1).

(1) He saw the _____(dancer/dancers).

Without additional context, such as a picture, this would be a confusing exercise given that both *dancer* and *dancers* are grammatically correct. Given this potential difficulty, we do not select tokens for which number is grammatically ambiguous.

Distractor generation After selecting targets that are unambiguously singular or plural, generating distractors is very straightforward. Let us assume that a given target ковёр *kovër* 'rug' results in the two morphological analyses in (2).

(2) a. ковёр+N+Msc+Inan+Sg+Nom
b. ковёр+N+Msc+Inan+Sg+Acc

To generate the distractors, we strip the case tag and generate all six cases from that base by adding the tags (+Nom, +Acc, +Gen, +Loc, +Dat, +Ins). For the example at hand, this generates the following respective forms: ковёр *kovër*, ковёр *kovër*, ковра *kovra*, ковре *kovre*, ковру *kovru*, ковром *kovrom*. Because the original token was singular, all of the generated wordforms are also singular. It is worth noting that generating wordforms in this way would not be possible with stochastic part-of-speech taggers, such as those developed by Sharoff et al. (2008a).

The generated forms are combined with the original token, and a set of unique wordforms is supplied to the learners as options in the multiple-choice activity. Currently, all six cases are used as distractors every time, but insights from SLA theory and future research should make it possible to identify those subsets of distractors most facilitating learning given a specific target.

Feedback Feedback for noun declension activities can be based on dependency relations established by a native-language syntactic parser. For example, in the phrase *On obyčno sidel rjadom s mamoj* 'He usually sat next to (his) mother.INS', the word *mamoj* is in the instrumental case because it is the object of the preposition *s*. This fact is explicitly represented in a dependency tree, since the preposition *s* directly dominates *mamoj*. The ATICALL system can consult the parse tree to prepare relevant feedback. If a learner selects the wrong case for this target, then the preposition *s* is highlighted to show the learner why it should be in instrumental. As in tutoring systems, miniature lessons could be prepared for specific syntactic constructions to provide related information. For example, with this preposition, the learner could be presented with the following: "*s* can govern three different cases depending on its meanings: INS='with', GEN='(down) from', and ACC='approximately'. (Use with ACC is rare.)"

This type of feedback is relevant, informative, and can easily be linked to specific syntactic constructions. Effective adaptive feedback in such a multiple-choice activity thus does not depend on learner-language NLP. The native-language NLP—both syntactic analyses and distractor generation—is providing effective feedback capabilities.

5.2.2 Stress

Five out of 10 of the Russian vowel letters are pronounced differently, dependent on stress position, namely а, о, я, е, and ё. Learners must know the stress position in order to know how to pronounce a written word. However, Russian stress patterns are specified lexically and cannot be predicted reliably from stem shape. Furthermore, many high-frequency lexemes have complex patterns of shifting stress, which means that many homographic forms of the same lexeme have different stress positions. This makes mastering the correct pronunciation of some words a difficult task for learners.

Four different activities were developed for stress. Unlike most 'highlight' activities in VIEW, the stress highlight activity does not make use of color, but simply adds a stress mark above every known stressed vowel in the text. For the 'click' activity, every vowel in the text becomes clickable: stressed vowels turn green and receive a stress mark; unstressed vowels turn red. The 'multiple-choice' activity selects some targets and learners try to identify the correctly stressed variant. The conventional use of the 'practice' activity is not well motivated for stress, since the entire set of possible responses is already represented in the 'multiple-choice' activity. Furthermore, typing stress marks is cumbersome for most users. Because of this, the 'practice' activity was replaced by an activity in which stressed vowels are highlighted when the cursor hovers over the token. In this section, we discuss

issues of target selection and distractor generation for the multiple-choice activity.

Target selection For many tokens, our constraint grammar is unable to completely disambiguate all of the readings of a given token. In such cases, the token can still be targeted if the remaining morphological ambiguity is immaterial with regard to stress. For example, the token *guby* can have three difference readings, given in (3), below. The nominative and accusative forms in (3-a) have the same stress position (on the stem), and the genitive form in (3-b) has the stress on the ending. If, for the sake of explanation, the constraint grammar removed the genitive reading, then the token could be used as a target even with remaining morphosyntactic ambiguity, since both of the remaining readings would have the same stress position. The fact that the form stressed on the first syllable in (3-a) is ambiguous between accusative or nominative is not relevant for our purposes; what matters is that it can be distinguished from the genitive form in (3-b).

(3) a. гу́бы
gúby
губа+N+Fem+Inan+Pl+Nom or +N+Fem+Inan+Pl+Acc
b. губы́
gubý
губа+N+Fem+Inan+Sg+Gen

Choosing targets for multiple-choice and practice activities is an interesting pedagogical issue, since almost every multisyllabic token is a potential target. Although there are many high-frequency words with difficult stress patterns, the overwhelming majority of Russian words have fixed stress. This means that if the program randomly selects targets for the multiple-choice and practice activities, many of the targets will not be pedagogically effective.

Stress patterns in Russian are specified lexically, and our solution to the target selection problem is also lexical. We compiled a stress activity target list consisting of lemmas that have shifting stress based on our morphological analyzer (chapter 2). For nouns, this includes Zaliznjak's stress indices *c*, *d*, *e*, and *f*. In addition, we also include masculine nouns with index *b* (end-stressed), such as *kón'* 'stallion'. In theory, end-stressed masculine nouns have fixed stress on the ending, but because the nominative singular has no ending, the stress falls on the stem, which means that at a surface level, the stress position does move. For adjectives, only short-form adjectives are targeted, since long-form adjectives do not ever have shifting stress. We also target one other large class of words: cognate words whose stress position in different in English and Russian. For example, compare English *rádiator* and Russian *radiátor*.

Proper nouns pose special problems for stress placement. Those proper nouns for in which stress position can vary by referent—such as the surnames *Iványov* and *Ivanóv*—are not targeted because identifying the referent cannot be achieved reliably with existing technology. However, proper nouns for which a single standard stress position can be defined—such as *Rossíja* 'Russia' or *Ukraína* 'Ukraine'—are added to the stress activity target list.

Distractor generation Generating distractors for the multiple-choice stress activity is currently done by simply giving every possible stress position. For words with four or fewer syllables, this approach is very reasonable, but for longer words, it may not be ideal to have so many options to choose from. For longer words, distractors would ideally mimic likely incorrect responses that learners would make on a parallel cloze test. In other words, the distractors should represent the kinds of mistakes that learners typically make. More research is needed to determine, but one possible source of information could be gained by logging user interaction with the system. It is possible that an analysis of learners' incorrect responses might yield patterns that could be exploited for generating more focused distractors, especially for longer words.

Feedback In the multiple-choice and practice activities for stress, targets are selected according to the stress activity target list introduced above, which is extracted from the morphological analyzer's source files, which are in turn based on Zaliznjak (1977). In Zaliznjak's dictionary, every word is assigned a code signifying which stress pattern it belongs to. We combined this information with frequency data from the Russian National Corpus in order to select an exemplar for each stress type. Based on this information, a tooltip is displayed that shows the exemplar and its paradigm when a learner gives an incorrect response. In this way, the learner is able to associate the targeted token with a word that is hopefully more familiar. This type of feedback supports both top-down and bottom-up learning, since it relies on an abstract connection to a concrete example.

5.2.3 Aspect

Most Russian verbs are either imperfective or perfective. For example, the English verb 'to say/tell' corresponds to the two Russian verbs *govorit'* (impf) and *skazat'* (perf). Imperfective verbs are generally used to express duration, process, or repetition. Perfective verbs are generally used for unique events, and they typically imply completion. An imperfective verb and a perfective verb that share the same lexical meaning are referred to as an aspectual pair. The choice of whether to use

one aspect or the other is frequently dependent on context, as we discuss in more detail in a corpus study below.

Russian has a productive system of aspectual derivation, by which so-called aspectual pairs are formed. Although some verb pairs have no derivational relation (like *govorit'* / *skazat'*), most verb pairs have one of the following two relations.[8] Example (4-a) exhibits an aspectual pair in which the perfective verb is decomposable as a prefixed form of the imperfective. Example (4-b) exhibits an aspectual pair in which the imperfective verb can theoretically be decomposed as a suffixed form of the perfective verb.

(4) a. IMPF: simplex verb ; PERF: prefix + simplex verb
smotret 'to watch.IMPF' / *po-smotret* 'to watch.PERF'
b. IMPF: (perfective stem) + suffix ; PERF: prefix + simplex verb
(ras-smatr)-ivat 'to examine.IMPF' / *ras-smotret* 'to examine.PERF'

Verbal aspect is arguably the single most challenging grammar topic for learners of Russian. The distinction between imperfective and perfective verbs is difficult for beginners to grasp, and even very advanced learners struggle to master the finer points. A set of ATICALL activities on aspect enables learners to focus on how aspect is used in context, which is crucial for mastering Russian.

Target selection Since aspect in Russian is lexical, target selection also takes a lexical approach. One problem in selecting targets for a multiple-choice activity is that not all verbs belong to an aspectual pair, which makes generating distractors problematic. For instance, the perfective verb *očutit'sja* 'find oneself at a location' does not have an imperfective counterpart. Since distractors should ideally be equivalent in every respect other than aspect, we select only verbs that belong to an aspectual pair.[9] The list of paired verbs is compiled from three sources: 1) pairings such as (4-a) above are taken from the Exploring Emptiness database[10], 2) pairings such as (4-b) above are taken from Zaliznjak (1977), and 3) pairings without a derivational relationship (of which there are few) are extracted from electronic dictionaries.

Choice of verbal aspect is generally a matter of construal, i.e., how the speaker is structuring the discourse, and some verb tokens could be grammatically correct with either aspect. Consider the English examples *John saw Mary* and *John had*

[8]For a more complete discussion of aspectual derivation, see Janda and Lyashevskaya (2011).

[9]The notion of aspectual pairs has been shown to be somewhat problematic by Kuznetsova (2013), who proposes that many verbs actually form aspectual clusters. However, from a pedagogical point of view, the concept of an aspectual pair is itself robust enough to serve as the basis for learning materials.

[10]`http://emptyprefixes.uit.no`

seen Mary. Even though they are likely to be used in different circumstances, both sentences are grammatically well-formed. Likewise, in Russian there are cases that allow either aspect. Meurers et al. (2010) suggested that lexical cues for English aspect and tense could be automatically identified by NLP. Indeed, many Russian textbooks and grammars also indicate contexts in which one aspect or the other is impossible, or at least very unlikely. In order to identify contexts which constrain the expression of one aspect or the other, Russian grammar books were consulted, resulting in the following lexical cues.

(5) Contexts in which perfective aspect is impossible/unlikely:
   a. Infinitive complement of *byt'* 'to be' (analytic future construction)
   b. Infinitive complement of certain verbs (especially phrasal verbs, such as 'begin', 'continue', 'finish', etc.)
   c. With certain adverbials denoting duration and repetition

(6) Contexts in which imperfective aspect is impossible/unlikely:
   a. Infinitive complement of certain verbs (e.g., 'forget' and 'succeed')
   b. With certain adverbials denoting unexpectedness, immediacy, etc.

A corpus study was conducted to test the usefulness of these features in an ATICALL application. The goal of the study was to determine the precision of the features, as well as their coverage, or recall. Precision was calculated as the percentage of verbs found adjacent to the appropriate lexical cues listed in (5) and (6) whose aspect was accurately predicted by that lexical cue. Recall was calculated as the percentage of all verbs whose aspect is correctly predicted by an adjacent lexical cue. In practical terms, for the purposes of our Russian aspect activities, precision tells us whether the learner ought to know which aspect is required, which is useful for target selection. Recall tells us what percentage of verbs actually appear together with these lexical cues, and whose aspect is correctly predicted by them.

The study included two corpora, each investigated separately. The Russian National Corpus[11] (230 M tokens) is a tagged corpus with diverse genres. The annotation in the RNC frequently contains ambiguities, but since the aspect of Russian verbs is rarely ambiguous and the aspect of the contextual features is irrelevant, ambiguous readings should not significantly affect our outcomes. Since the RNC does not include syntactic relations, we rely on collocation of these lexical cues with verbs. SynTagRus[12] (860 K tokens) is a morphologically disambiguated and syntactically annotated dependency treebank of Russian. Because dependency relations are defined, identifying adverbial relations and verbal complements is

[11] http://www.ruscorpora.ru/en/index.html
[12] http://www.ruscorpora.ru/search-syntax.html

straightforward. The results are given in Table 5.1.

	RNC	SynTagRus
Precision	0.95	0.98
Recall	0.03	0.02

Table 5.1: Results of the corpus study of lexical cues for aspect

The precision of these lexical cues is very high, meaning that when lexical cues are present, the verb is of the predicted aspect. This is expected, since known counterexamples such as (7) are uncommon.

(7) Настоящий друг всегда скажет правду.
Nastojaščij drug vsegda skažet pravdu.
True friend always will-tell.PF truth
'A true friend will always tell the truth.'

Given that Russian allows variable word order, it is surprising that collocation in the RNC is nearly as reliable as dependency relations in this task. Apparently these lexical cues have a very strong tendency to appear adjacent to the verbs that they modify or are in a construction with.

Unfortunately the recall of the lexical cues is extremely low. It correctly predicted the aspect of only one out of 50–60 verbs. Although future work is needed to explore these phenomena more thoroughly, these results seem to indicate that verbal aspect in Russian is predominantly determined suprasententially, with lexical cues playing only a very minor role.

For language learning, this result has several implications. First, it shows that learners can place their confidence in lexical cues, but these cues will not get them very far. Yet in Russian textbooks, more space is often dedicated to these lexical cues than to discourse considerations. This means that some learners may not be getting enough instruction on strategies that help in the majority of cases. Second, for the purposes of target selection, the Russian VIEW system can rely on lexical cues of aspect with some confidence. If a token is adjacent to the appropriate cues, then a learner should be expected to know the aspect of that token. However, since the lexical cues are so sparse, the system cannot make an intelligent decision for the overwhelming majority of verb tokens. One potential solution would be to implement machine-learning approaches to predict the distribution of each aspect more accurately. However, even though such models might make more accurate predictions, there is no guarantee that its output would reflect what a human second language learner should be capable of distinguishing.

If it is true that context-based rules cannot provide adequate coverage of as-

pectual usage, then this implies that Russian verbal aspect is acquired through semantic bootstrapping. As learners are exposed to verbs of both aspects, real-world knowledge and expectations form the foundation upon which aspectual categories are built in their minds. Therefore, it may not be feasible for an ATICALL system to predict how or whether the learner can be expected to know the aspect of a given target. However, this does not mean that the system cannot provide a significant benefit to the learner by facilitating focus-on-form exercises, albeit blindly. For now, our system selects any paired verbs as targets, giving preference to forms that appear adjacent to our lexical cues.

Distractor generation Distractors for the multiple-choice activity are generated by replacing the lemma with its aspectual partner, and replacing the aspectual tag, as shown in (8).

(8) a. Original: читать+V**+Impf**+TV+Pst+Msc+Sg
b. Distractor: прочитать+V**+Perf**+TV+Pst+Msc+Sg

Feedback As we discussed in above, determining *why* a given aspect is required in a given context is rarely possible with current technology. However, some tokens do have a clear lexical cue, which is used both to promote their selection as targets, and can also be used as corrective feedback. For example, given the sentence *On obyčno sidel rjadom s mamoj.* 'He usually sat.IMPF next to (his) mother', if the learner selects the perfective verb, then the adverb cue *obyčno* can be highlighted to show the learner *why* perfective is not appropriate. The information needed to give enhanced feedback is the same information used in target selection.

5.2.4 Participles

Russian has four kinds of adjectival participles, which are used both attributively and as relativizers. Their formation, meaning, and usage are not usually introduced to learners until more advanced levels. Although they are not used frequently in spoken Russian, participles are very common in written Russian, especially in high registers, such as literature, official documents, news, and technical writing. In these domains, participles figure prominently in the structure of complex sentences. Many learners without parallel forms in their L1 struggle with Russian participles. All of these things make participles an excellent candidate for ATICALL visual input enhancement.

Target selection The four participles are present active, present passive, past active, and past passive. The passive participles are generally only formed from

transitive verbs. Present participles are only formed from imperfective verbs, and past participles are typically formed from perfective verbs. The result of this is that not all verbs (or rather, verb pairs) can form every kind of participle. In order to select only those verbs from which a full 'paradigm' of distractors can be formed, we limit target selection to transitive verbs that are members of aspectual pairs (as described in section 5.2.3). We also do not target participles that have a possible lexicalized adjective reading, such as одетый *odetyj* 'dressed', or participles in the short-form.

Prompt generation Multiple-choice and cloze activities require a prompt for learners to know which kind of participle is being elicited. One way to do this is to rephrase the participle using the relative determiner который *kotoryj* 'which'. For example, the present active participle дремлющий *dremljuščij* 'slumbering' can be rephrased as который дремлет *kotoryj dremlet* 'which/who slumbers'. Fortunately, it is possible to perform this rephrasing automatically, based solely on the tags of the original token. We refer to the resulting paraphrase as a *relative-rephrase*. This is demonstrated in (9) and (10), where (a) gives an example of a participle in context, (b) gives the participle's grammar tags assigned by the tagger, and (c) provides the *relative-rephrase* and its readings. The bolded tags in (b) and (c) indicate the tags that are extracted from the participle reading in order to generate the relative-rephrase. The tags in (c) that are not bolded are the same for every participle of that category.

(9) Present Active

a. разлука есть гроб, заключающий в себе половину сердца
separation is tomb which-imprisons in itself half of-heart
'separation is a tomb which imprisons half of one's heart.'

b. заключающий: заключать+V+Impf**+TV+Prs**Act**+Msc+Sg**+Nom

c. который заключает 'which imprisons'
который+Pron+Rel**+Msc+Sg**+Nom
заключать+V+Impf**+TV+Prs+Sg**3

(10) Past Passive

a. Рассеянное молчание
which-was-scattered silence
'scattered silence'

b. рассеять+V**+Perf**+TV**+Pst**Pss**+Neu+Sg**+Nom

c. которое рассеяли 'which (they) scattered'
который+Pron+Rel**+Neu+Sg**+Acc
рассеять+V**+Perf**+TV**+Pst**+MFN+Pl

The given relative-rephrasing of passive participle in (10) is a zero person construction (неопределённо-личное предложение), in which there is no explicit subject, and the verb shows third-person plural agreement. Although this rephrasing is not always the best possible rewording of the passive, it is the alternative that works best in a wide variety of circumstances.

This method of prompt generation takes advantage of the systematicity of grammatical relations in Russian. It works because all of the morphosyntactic information needed to form the relative-rephrase is already present in the original participle's morphosyntactic tags.

Feedback Recall that the participle activities discussed above have a prompt provided in the form of a *kotoryj* 'which/who' relative-rephrase of the participle. It was shown that the morphosyntactic properties of the participle correspond directly to the morphosyntactic properties of the relative-rephrase. These very same relations can be leveraged to provide feedback to the learner.

For example, let us say that the original token was a past active participle *napisavšij* 'who wrote' with the relative-rephrase hint (*kotoryj napisal*). If the learner selects the present active participle distractor *pišuščij*, they could be presented with feedback such as: "The word you selected means *kotoryj pišet*. Pay attention to the tense of *napisal*." This feedback is tailored to the learner's response, and encourages the learner to compare the functional meanings of the relevant morphological forms. In this case, the strategy used for prompt generation facilitates customized feedback.

5.3 Feedback

Overall, the four examples sketched above show that the provision of specific types of adaptive feedback is a meaningful and natural extension of an ATICALL system such as VIEW, using the same NLP techniques employed in analysis, target selection and distractor generation. However, the nature of the feedback based on native-language NLP can potentially differ from feedback based on learner-language NLP.

Since the publication of Truscott (1996), which claims that grammar correction in L2 writing impedes learning, several researchers have investigated the effects of feedback, with many studies finding a useful distinction between *indirect*, *metalinguistic*, and *indirect* feedback (Ellis, 2009). Indirect feedback is when the teacher indicates that an error was made, but does not actually correct it. Metalinguistic feedback is when the teacher indicates that a mistake was made, together with some kind of explicit comment about the nature of the error. Direct feedback is when the

teacher provides the learner with the correct form. The distinction between the different kinds of corrective feedback aligns well with the different kinds of information that are available from native-language and learner-language NLP.

From the standpoint of ATICALL activities (with native-language NLP), both indirect and direct feedback are straightforward to generate automatically, since the correct form is known to the system by virtue of being the original text from which the exercise was generated. If a learner's response does not match the form in the original text, the ATICALL system can simply indicate that the response is incorrect (indirect feedback), or show the presumed correct form from the original text (direct feedback). However, as we have demonstrated above, native-language NLP makes it possible to give metalinguistic feedback, based on the morphological and syntactic analyses of the original text.

One the other hand, intelligent language tutoring systems (with learner-language NLP) are well attuned to learner errors, since the NLP is primarily designed to process and diagnose errors typical of language learners. Just as with ATICALL applications, an ILTS can give indirect feedback to incorrect responses by simply indicating that there is an error. An ILTS can also give metalinguistic feedback, based on the diagnostic abilities of its learner-language NLP. For example, a Russian learner may type the word *pišjut* instead of *pišut* '(they) write', forgetting to apply a spelling rule that *ju* changes to *u* after *š*. The learner-language NLP would typically diagnose this error as a spelling-rule error, and can give metalinguistic feedback based on the that information. In the case of direct feedback, an ILTS must wrestle with the additional complication of identifying what the correct form should be in the first place, but assuming that it can do so successfully, an ILTS can give direct feedback as well.

To summarize, both ATICALL and ILTS can give both indirect and direct feedback to learners, although in the case of direct feedback, an ILTS must overcome the challenge of determining what the correct form should be. However, with regard to metalinguistic feedback, the kinds of feedback that can be facilitated by native-language NLP and learner-language NLP are qualitatively different. Native-language NLP can facilitate metalinguistic feedback that describes the expected correct response, whereas learner-language NLP is generally more attuned to providing metalinguistic feedback that is focused on the kind of error the learner made. This distinction is one that, to our knowledge, has not been made in any studies of corrective feedback. On the contrary, most studies assume that metalinguistic feedback is error-focused (e.g. Lyster and Ranta, 1997; Ellis, 2009). Given the controversy that currently exists over the effectiveness of different kinds of corrective feedback, future research may benefit from recognizing the distinction we have identified in two separate categories of metalinguistic corrective feedback.

5.4 Conclusions and Outlook

We reported practical and theoretical issues related to developing automatic visual input enhancement for Russian, with a focus on including adaptive feedback in such an ATICALL system. The selected topics demonstrate the challenges that a morphology-rich language brings with it and how a rule-based morphological analysis can be used to tackle them. In addition to providing the means for effective disambiguation, the finite-state approach makes it possible to generate wordforms for distractors, prompts (participles), and stressed wordforms. It also makes it possible to employ an epistemic/intuitionistic logic to select target tokens based on both practical and pedagogical considerations. The system is self-aware, in a sense, and knows whether it knows enough information about each token. For example, tokens for which the constraint grammar could not sufficiently disambiguate morphosyntactic readings can be avoided. Furthermore, where possible, targets are selected according to their pedagogical value, rather than purely random selection.

We also characterized certain types of adaptive feedback, which typically associated with intelligent language tutoring systems, that can be added in an ATICALL environment using the same information that is used for target selection and distractor generation. This refines the perspective distinguishing two subdisciplines of ICALL (Meurers, 2012), while keeping a clear distinction on the processing side between analyzing learner language and analyzing native language for learners.

In terms of future work, the crucial next step is to empirically evaluate the approach and the specific parameterization (activities, enhancement methods, distractors, and feedback used) in terms of learner uptake and attitudes, and more generally, learning gains. While identifying a real-life educational context in which the tool can be integrated meaningfully is a complex undertaking, the computational approach presented in this chapter should readily support a controlled study with different intervention groups and a standard pretest-posttest-delayed posttest design. The foundational hypotheses upon which visual input enhancement is built have not been empirically evaluated to a sufficient degree (Lee and Huang, 2008), so evaluating learner outcomes is needed not only to establish the system's effectiveness, but also to validate the theories upon which it is based. As already suggested in Meurers et al. (2010), an ATICALL platform such as VIEW should make it possible to push intervention studies to a level where effects could be more readily established than in the very controlled but small laboratory settings.[13] This seems particularly relevant since there are many parameters that need to be explored, e.g., which kind of visual input enhancement works for which kind of learners and for which kind of linguistic targets presented in which contexts. We

[13] In a similar vein, Presson et al. (2013) discuss the potential of experimental computer-assisted language learning tools for SLA research.

are also interested in exploring which kind of distractors (and how many) are optimal for which activities or learner levels. Finally, while it is beyond the current analysis we perform, we plan to investigate different ways of measuring *noticing* through computer interaction behaviors, and test their correlation with individual learner characteristics and learning outcomes.

On the computational side, although we have evaluated the performance of the NLP components used in the approach in terms of precision, recall, and speed (Tyers and Reynolds, 2015), we believe that it is also important to evaluate its performance for the specific parts of speech and morphological properties that are at issue in a given activity. For the activities discussed in this chapter, this includes nouns for the noun declension activity, infinitive and indicative verbs for the aspect activity, participles for the participles activity, and all parts of speech for the stress activity. The performance should also be tested on different genres and reading levels, since those distinctions will affect NLP performance. Ideally, the performance should be analyzed on a corpus that is characteristic of the material that the learners or their teachers select as the basis for generating activities – which is only possible in an interdisciplinary approach including both NLP research and real-life teaching and learning contexts.

In terms of making an ATICALL system useful in real life, an important challenge arises from the fact that many texts do not contain enough of the relevant sorts of targets or contextual cues. This, for example, was apparent in the corpus study related to verbal aspect. The texts a learner chooses for enhancement and activity generation thus should be filtered in a way ensuring a sufficient number of targets in the texts. To address that need, we plan to further develop grammar-aware search engines (Ott and Meurers, 2010) supporting the selection of appropriate materials. The first steps toward this end are presented in the following chapter, in which I present research on automatically classifying Russian texts according to L2 readability levels.

Chapter 6

Automatic classification of document readability on the basis of morphological analysis

6.1 Introduction

Reading is one of the core skills in both first and second language learning, and it is arguably the most important means of accessing information in the modern world. Modern second language pedagogy typically includes reading as a major component of foreign language instruction. Over the years, there has been some debate regarding the use of authentic materials versus contrived materials, where *authentic* materials are defined as "A stretch of real language, produced by a real speaker or writer for a real audience and designed to convey a real message of some sort" (Morrow, 1977, p. 13).[1] Many empirical studies have demonstrated advantages to using authentic materials, including increased linguistic, pragmatic, and discourse competence (Gilmore, 2007, citations in §3). However, Gilmore notes that "Finding appropriate authentic texts and designing tasks for them can, in itself, be an extremely time-consuming process." Finding appropriate texts is difficult because "appropriate" has several important connotations. An appropriate text should arguably be interesting, linguistically relevant, authentic, recent, and at the appropriate reading level.

Sometimes teachers select texts only from books or collections that are known to be at the appropriate reading level. This frequently results in texts that are not relevant, interesting, or current. Another strategy is to search on the internet for

[1] The definition of authenticity is itself a matter of disagreement (Gilmore, 2007, §2), but Morrow's definition is both well-accepted and objective.

texts on a relevant topic, selecting whichever text is closest to the appropriate reading level, or modifying a text to make it accessible. The second strategy is time-consuming and difficult, since most of the texts returned on an internet search query are inappropriate for beginning learners. Tools to automatically identify a given text's complexity would help remove one of the most time-consuming steps of text selection, allowing teachers to focus on pedagogical aspects of text selection. Furthermore, these tools would also make it possible for *learners* to find appropriate texts for themselves.

A thorough conceptual and historical overview of readability research can be found in Vajjala (2015, §2.2). The last decade has seen a rise in research on readability classification, primarily focused on English, but also including French, German, Italian, Portuguese, and Swedish (Roll et al., 2007; Vor der Brück et al., 2008; Aluisio et al., 2010; Francois and Watrin, 2011; Dell'Orletta et al., 2011; Hancke et al., 2012; Pilán et al., 2015). Broadly speaking, these languages have limited morphology in comparison with Russian, which has comparatively rich morphology among major world languages. It is therefore not surprising that morphology has received little attention in studies of automatic readability classification. One important exception is Hancke et al. (2012) which examines lexical, syntactic and morphological features with a two-level corpus of German magazine articles. In their study, morphological features are collectively the most predictive category of features. Furthermore, when combining feature categories in groups of two or three, the highest performing combinations included the morphology category, which I interpret to mean that the morphology category of features encodes information that is not found in other categories. If morphological features figure so prominently in German readability classification, then there is good reason to expect that they will be similarly informative for Russian second-language readability classification.

Studies of automatic readability assessment based on machine learning rely on training corpora consisting of texts that have been rated by humans, whether teachers, publishers, or students. Even for humans, readability assessment is a very difficult task. The fact that many popular word processors include automatic readability assessment tools indicates that humans are frequently unsure about how difficult their own writing is. Therefore, there is some reason to doubt the validity of human ratings in a readability corpus. However, without superior alternatives, most readability researchers use human ratings as the gold standard, without examining their validity.

This chapter explores to what extent textual features based on morphological analysis—as made available by the Russian morphological analyzer described in Chapters 2 and 3—can lead to successful readability classification of Russian texts for language learning. More specifically, I train classifiers to assign a text to one of

six CEFR proficiency levels: A1, A2, B1, B2, C1 and C2. I also examine the internal validity of the ratings in my gold standard corpus in order to estimate how well the resulting classifier models can be applied to new, unseen texts. In Section 6.2, I give an overview of previous research on readability, including some work on Russian. The corpora collected for use in this study are described in Section 6.3. The features extracted for machine learning are outlined in Section 6.4. Results are discussed in Section 6.5, and conclusions and outlook for future research are presented in Section 6.7.

6.2 Background

The history of empirical readability assessment began as early as 1880 (DuBay, 2006), with methods as simple as counting sentence length by hand. Today, research on readability is dominated by machine-learning approaches that automatically extract complex features based on surface wordforms, part-of-speech analysis, syntactic parses, and models of lexical difficulty. In this section, I give an abbreviated history of the various approaches to readability assessment, including the kinds of textual features that have received attention. Although some proprietary solutions are relevant here, I focus primarily on work that has resulted in publically available knowledge and resources.

6.2.1 History of evaluating text complexity

The earliest approaches to readability analysis consisted of developing readability formulas, which combined a small number of easily countable features, such as average sentence length, and average word length (Kincaid et al., 1975; Coleman and Liau, 1975). For example, the well-known Flesch-Kincaid Reading Grade formula is computed as:

$$206.835 - 1.015\left(\frac{total\ words}{total\ sentences}\right) - 84.6\left(\frac{total\ syllables}{total\ words}\right) \tag{6.1}$$

The constants in the formula are weights intended to yield a result that indicates the US grade level for which the text is appropriate. These weights have been adapted for specific genres and languages. Although formulas for computing readability have been criticized for being overly simplistic, they were quickly adopted and remain in widespread use today.[2] An early extension of these simple 'counting' formulas was to additionally rely on lists of words deemed "easy", primarily

[2]The Flesch Reading Ease test and the Flesch-Kincaid Grade Level test are implemented in the proofing tools of many major word processors.

based on their frequency, or polling of young learners (Dale and Chall, 1948; Chall and Dale, 1995; Stenner, 1996). A higher proportion of words belonging to these lists resulted in lower readability measures, and vice versa.

With the recent growth of natural language processing techniques, it has become possible to extract information about the lexical and/or syntactic structure of a text, and automatically train readability models using machine-learning techniques. Some of the earliest attempts at this built unigram language models based on American textbooks, and estimated a text's reading level by testing how well it was described by each unigram model (Si and Callan, 2001; Collins-Thompson and Callan, 2004). This approach was extended in the REAP project[3] to include a number of grammatical features as well (Heilman et al., 2007, 2008a,b).

Over time, readability researchers have increasingly taken inspiration from various subfields of linguistics to identify features for modeling readability, including syntax (Schwarm and Ostendorf, 2005; Petersen and Ostendorf, 2009), discourse (Feng, 2010; Feng et al., 2010), textual coherence (Graesser et al., 2004; Crossley et al., 2007a,b, 2008), and second language acquisition (Vajjala and Meurers, 2012). The present study expands this enterprise by examining the contribution of morphological features as a measure of textual complexity.

6.2.2 Automatic readability assessment of Russian texts

The history of readability assessment of Russian texts takes a very similar trajectory to the work related above. Early work was based on developing formulas based on simple countable features. For example, Mikk (1974) hypothesized that the more abstract a text is, the more difficult it is to understand. Mikk proposed counting the number of tokens that refer to abstract ideas, based primarily on derivational suffixes. Other researchers adapted readability formula for English to Russian by adjusting weights, or introducing other features. For instance, Oborneva (2005, 2006a,b) adapted the Flesch Reading Ease formula for the Russian language by means of adjustment coefficients. She compares the average length of syllables in English and Russian words and percentage of multi-syllable words in dictionaries for these languages. Mizernov and Graščenko (2015) compared 12 different traditional readability formulas, and created an application for comparing their output.

Some researchers have tried to be more objective about defining readability, by obtaining data from expert raters, or from other experimental means. For example, Petrova and Okladnikova (2009); Okladnikova (2010) studied the readability of testing materials. They performed a regression analysis of test items rated according to difficulty by expert raters against a number of features. Most of the features

[3] `http://reap.cs.cmu.edu`

discussed in the paper are not relevant for the present study, since they are specific to a testing domain, and therefore do not apply to running text.

Likewise, Špakovskij (2003, 2008) conducted a series of experiments in order to automatically evaluate the difficulty of texts in chemistry textbooks. Students were given texts from chemistry textbooks, along with exercises to test their understanding. The students were asked to rate the difficulty of the text; their performance on the exercises, and the time required to complete the entire task, were recorded. Eighty-three textual features were extracted—some manually and others automatically—based on lexical properties, morphological categories, typographic layout, and syntactic complexity. Based on the results from linear regression models using subsets of these features, three factors were judged to be the most informative as a group: percentage of words with more than 8 letters, percentage of tokens that were formal terminology, and the number of symbols in chemical reactions. These three factors were then combined in a formula using the weights from the regression models to predict the difficulty of textbook sections.

Ivanov (2013) calculated the Pearson product-moment correlation coefficient for 45 factors, using a corpus of 88 works of literature recommended for children between 4th and 11th grades. 10 factors yielded significant correlations: number of short-form adjectives, number of verbs in personal forms, Flesch-Kincaid Reading Level, Flesch Reading Ease for Russian (Oborneva, 2005, 2006a,b), number of subordinating conjunctions, Coleman-Liau Index, number of coordinating conjunctions, number of abstract words, and the number of pronouns.

Filippova, Krioni and Nikin (Nikin et al., 2007; Krioni et al., 2008; Filippova, 2010) published a number of works reporting on development of an application designed to help readers and authors to gain access to information derived from complexity measures of a given text. The author module is relevant to the current discussion. The program automatically extracts a number of features from the text and highlights/annotates the text so that the author can reduce the complexity of the text. These features were not evaluated against a gold standard corpus, but they propose some features that are worth consideration, as listed below:

1. number of definitions (identified by 10 constructions such as “X is Y”,“is called”, “means”, “represents”, “signifies”, etc.)

2. number of abstract words (14 derivational affixes)

3. number of sentences containing long words

4. number of participles and verbal adverbs

5. ratio of sentences containing participles and verbal adverbs

6. number of complex sentences (simple and complex conjunctions)

Kotlyarov (2015) reports preliminary results of some pilot testing of a small number of text complexity features inspired by Čukovskij (2009) and Gal' (2014), who offer guidelines for avoiding "official-ese", a tendency to overcomplicate texts that the writer deems important. These features include, semi-auxiliary verbs,[4] participles, verbal adverbs, chains of interdependent oblique noun phrases (especially long chains of genitives), passive voice, less common words, sentence length and number of subordinate clauses. Preliminary results showed highest correlation with "complex verb forms"[5] and chains of genitives. Kotlyarov does not clearly state what these features correlate with, but it appears that the Flesch Reading Ease score is taken as a gold standard.

Russian text complexity for language learners

The experiments reported in this chapter are concerned with text complexity for language learners. Although we can expect many of the factors of text complexity to be the same for both native speakers and learners, there are certainly other factors that differ, especially since learners generally have more limited vocabularies, as well as limited knowledge of the target languages' syntactic and morphological constructions. Only a handful of studies have investigated Russian readability for language learners, and they are briefly described below.

Wądołowska-Lesner (2011) investigated the effect of three factors on readability of Russian texts by native-Polish learners: syntactic complexity (measured as sentence length in words), lexical difficulty (measured as non-repeating words and words whose Polish translations are similar), and lexical complexity (measured as the number of tokens with 3 or more syllables). These data were collected for three texts and compared with both ratings and qualitative evaluations of 59 students who read the texts. Not surprisingly, their evaluations support the conclusion that all three factors are important contributors to readability.

Sharoff et al. (2008b) considered a number of features for comparison between two subcorpora: original texts published in quality online newspapers, and the Russian BBC website, which they judged to be significantly easier for native English students. They investigate 11 features: token frequency (top 1000/2000/3000 words), average sentence length, average word length in syllables, Flesch Reading Ease, coverage by more frequent part-of-speech trigrams, and the average number of conjunctions, verbs, passive verbs, modal verbs, prepositions, or punctuation

[4]Kotlyarov does not explain what he means by the term semi-auxiliary, which is not a standard term in Russian linguistics.

[5]By "complex verb forms", Kotlyarov seems to mean verbal adverbs and participles.

marks per sentence. A principle component analysis of these features supported the intuition that texts with a higher number of less frequent words, conjunctions, prepositions, and longer sentences tend to be more difficult for language learners.

Recently, Karpov et al. (2014) performed a series of experiments using several different kinds of machine-learning models to automatically classify Russian text complexity, as well as single-sentence complexity. They collected a small corpus of texts (described in Section 6.3.1 below), with texts at 4 of the CEFR levels:[6] A1, A2, B1, and C2. They extracted 25 features from these texts, including the following:

1. Document length in words
2. Document length in letters
3. Average sentence length in words
4. Average sentence length in syllables
5. Average sentence length in letters
6. Average word length in syllables
7. Average word length in letters
8. Percentage of words with N or more syllables ($3 \leq N \leq 6$)
9. Percentage of words with N or more letters ($5 \leq N \leq 13$)
10. Percentage of words not in active vocabulary of CEFR level A1, A2, or B1
11. Occurrence of part of speech X, where...
    - $X \in$ {NOUN, ADJF, ADJS, COMP, VERB, INFN, PRTF, PRTS, GRND, NUMR, ADVB, NPRO, PRED, PREP, CONJ, PRCL, INTJ}

Using these features in Classification Tree, SVM, and Logistic Regression models for binary classification (A1-C2, A2-C2, and B1-C2), they report achieving accuracy close to 100%. It is not clear why Karpov et al. divide their binary classifiers in this way, as opposed to stepwise combinations, such as A1-A2, A2-B1, and B1-C2. There is no real utility in making binary distinctions that skip intervening levels. In a four-way classication task, they report that their results were lower, but they only provide precision, recall, and accuracy metrics for the B1 readability level during four-way classification, which were as high as 99%. Despite the

[6] CEFR levels are introduced in Section 6.3.

strangely selective reporting in their article, I made efforts to replicate their results. However, the copy of the corpus that I received from them has about half as many documents at the A1 level as are reported in the paper. The authors could not provide an explanation for the discrepancy. With the given data, my attempts at replicating their study yielded lower results.

Karpov et al. (2014) also performed an Information Gain Ratio analysis, revealing that the top-ten most informative features were the three features regarding vocabulary coverage, followed by percentage of words with 7/8/9/10 or more letters, as well as average sentence length in syllables, average word length in letters, and percentage of words with 5 or more syllables. Notably, the features based on part of speech, which is the closest they have to morphology, are missing from the top ten.

Summary Section 6.2.2 has presented an overview of the published research connected with automatic assessment of Russian text complexity. Although many of these studies investigated textual properties that can serve as inspiration for the present study, only one other study (Karpov et al., 2014) has investigated features for classifying Russian texts according to L2 reading levels. Surprisingly, that study included virtually no morphological features. Karpov et al. (2014) was limited by the fact that it covered only four out of six CEFR levels with no more than 60 data points per level. Furthermore, irregularities in their reporting make it difficult to draw firm conclusions from their work.

6.3 Corpora

All of the corpora used in this study are based on the same scale for rating readability, the Common European Framework of Reference for Languages (CEFR). The six common reference levels of CEFR can be divided into three levels – Basic user (A), Independent user (B), and Proficient user (C)—each of which is subdivided into two levels—1 and 2. This yields the following six levels in ascending order: A1, A2, B1, B2, C1, and C2.[7]

Multiple corpora were used in this study, each of which is described in its own subsection below. In addition to increasing the number of data points for machine learning, using a variety of corpora has one important benefit. Assuming that the criteria used to determine the readability of texts differ between corpora, the likelihood of overfitting can be reduced by training on more than one corpus. On the other hand, it opens the possibility that each corpus' readability ratings

[7]There is no consensus on how the CEFR levels align with other language evaluation scales, such as the ACTFL and ILR used in the United States.

are not well aligned; one corpus' B1 rating might be closer to another corpus' B2 rating, etc. However, the validity of each corpus' ratings can be confirmed by comparing the output of models trained on various subsets of the corpora. This is especially important for corpora from less authoratative sources.

In the following sections, I give a brief overview of the sources and profiles of each of the corpora used in this study, the final section giving a summary of all the corpora together. Some of the corpora used in this study are proprietary, and so they cannot be published online. However, they can be shared privately for research purposes, and I welcome any such inquiries. With the exception of the two corpora taken from Karpov et al. (2014), all of the corpora were created and used for the first time in this study.

Many of the texts in these corpora come from pedagogical sources that include 'back matter' (i.e. glosses, questions, exercises, and attributions). Several of the machine-learning features discussed below would be affected significantly by including the back matter in the corpus. For example, the back matter is very likely to repeat tokens from the main text, which would affect features like the type-token ratio. For this reason, the back matter was not included in the corpus.

6.3.1 CIE corpus

The authors of Karpov et al. (2014) were kind enough to share with me the corpus used in their study. The corpus contains 195 documents, which can be subdivided into two subcorpora. The first subcorpus includes texts created by teachers for learners of Russian. These texts are taken from a collection of materials kept in an open repository at `http://texts.cie.ru`, which is maintained by the Center for International Education at Lomonosov Moscow State University. There are 28 texts at level A1, 57 texts at level A2, and 60 texts at level B1. Note that Karpov et al. report having 52 documents at the A1 level, whereas the corpus I received from them contains only 28. The repository at `texts.cie.ru` also contains only 28 documents at the A1 level. The authors were unable to provide a reason for this discrepancy.

6.3.2 news corpus

The second subcorpus used by Karpov et al. (2014) consists of 50 original news articles for native readers. These texts were all rated as level C2, although it is unclear to what extent these news articles were checked by hand to confirm their reading difficulty.

6.3.3 LingQ corpus

The LingQ corpus is a corpus of texts from `http://www.lingq.com`, a commercial language-learning website that includes lessons uploaded by member enthusiasts. Using the website's Application Program Interface, I downloaded all 3744 Russian lessons, after which each lesson was cleaned or removed by hand. The decision to delete all or part of a lesson's text was ultimately subjective, but I based my decisions on the following criteria. First, authorship attributions and dates, such as "(написано и прочитано Евгением40, 2015) 'written and read by Evgenij40, 2015', were removed. Second, texts that consisted of lists of vocabulary, phrases, incomplete sentences, or fragments were deleted. Third, all back matter, such as glosses, exercises, or questions about the text were removed. Fourth, meta-linguistic explanations or commentaries were removed. Fifth, some of the texts were clearly copy/pasted from other resources, leaving footnote or other artifacts, such as '[1]'. These artifacts were removed. Lastly– and most subjectively –any text that seemed like it could not reasonably be used as a text for classroom or individual study was discarded.

The final corpus contains 3481 texts: 323 at level A1, 653 at A2, 716 at B1, 832 at B2, 609 at C1, and 348 at C2. The reading level of each text was determined by the member who uploaded each lesson, so the validity of the ratings is dependent upon the expertise of the person who posted the lesson. Because there were possible discrepancies between how members rated their texts, I created a subcorpus of only those texts that were uploaded and rated by members who uploaded at least 50 lessons spread across at least three different reading levels. I refer to these contributors as 'experts'. Table 6.1 gives a summary of the contributions of those who met these criteria.

username	levels	courses	lessons	notes
evgueny40	6	62	1585	Native Russian, professional teacher
LingQ_Support	6	76	645	official website contributor(s?)
Ress	6	17	155	no profile
mikola	5	11	90	no profile
Solena	4	4	60	Native Russian
Polk00	4	4	59	Native Russian
MissTake	3	13	444	Native French
keke_eo	3	5	124	Native English
lomsa	3	16	107	Native Russian

Table 6.1: Contributions of LingQ 'expert' Russian contributors

The first two contributors listed in Table 6.1 together represent more than half

of all the Russian lessons on LingQ, and both contributed to all six reading levels. Because their contributions were significantly higher than the other experts, I separated each of them into unique subcorpora, which I will refer to as 'Evgenij' and 'LQsupp'. This results in four LingQ subcorpora: Evgenij (contributions of evgueny40), LQsupp (contributions of LingQ_support), Expert (contributions of remaining experts listed in Table 6.1), and lq (contributions of the remaining 54 non-experts). The distribution of each LingQ subcorpus across levels is given in Table 6.2.

username	Total	A1	A2	B1	B2	C1	C2
Evgenij	1387	169	516	446	173	58	25
LQsupp	618	61	42	51	292	106	66
Expert	1021	73	68	159	298	387	36
lq	455	20	27	60	69	58	221
Total	3481	323	653	716	832	609	348

Table 6.2: LingQ subcorpora distribution of documents by level

The words per document for each of these subcorpora is given in Table 6.3. In general, document length increases with higher CEFR levels, but there is a surprising trend for C2 texts to be shorter than C1 texts.

username	Total	A1	A2	B1	B2	C1	C2
Evgenij	267	60	56	295	581	1376	779
LQsupp	1692	102	150	285	2797	1288	988
Expert	2130	74	151	262	352	5036	1771
lq	645	151	210	438	775	1825	448
Total	1116	77	78	299	1293	3729	711

Table 6.3: LingQ subcorpora distribution of words per document by level

6.3.4 Red Kalinka corpus (RK)

The Red Kalinka corpus is a collection of 99 texts taken from 13 books in the "Russian books with audio" series available at `http://www.redkalinka.com`. These books include stories, dialogues, texts about Russian culture, and business dialogues. There are 40 texts at level A1, 18 texts at level A2, 17 texts at level B1, 18 texts at level B2, 6 texts at level C1, and no texts and level C2.

Texts were extracted from the original `pdf` files and preprocessed to remove artifacts of the `pdf` format, such as missing whitespace, erratic placement of stress marks, converting combining diacritics to standard cyrillic characters, etc.

6.3.5 TORFL corpus

The Test of Russian as a Foreign Language (TORFL) is a set of standardized tests administered by the Russian Ministry of Education and Science. Passing the test at certain levels qualifies an individual for citizenship, entrance into institutions of higher education, receipt of college degrees, work in language-oriented professions, and receipt of college degrees in fields related to the Russian language. Each of level is clearly defined with particular competencies and required vocabulary. Because of the rigor with which these texts were created, the validity of their ratings can be expected to be the strongest of the corpora used in this study.

Each TORFL test includes sections directed at reading comprehension, with texts and questions. The TORFL corpus is a collection of such texts that I extracted from official practice tests for the TORFL. The corpus contains 168 texts, fairly evenly distributed across all six CEFR levels: 31 at level A1, and 36, 36, 26, 28, and 11 at the subsequent levels.

6.3.6 Zlatoust corpus (Zlat.)

The Zlatoust Publishing House, which has exclusive rights to publishing official materials for the TORFL, also publishes a series of readers for language learners at the lower CEFR levels. The Zlatoust corpus (sometimes abbreviated as Zlat, below) is a collection of texts extracted from a large portion of the books in this series of readers. Each book has been assigned to a given reading level, and I make the assumption that all of the chapters or sections within a given book are of the same reading level. This assumption seems to be warranted, since the books are clearly designed to be read one chapter at a time. With 746 documents, the Zlatoust corpus is the second largest corpus included in this study. The distribution between reading levels is very uneven, with 66 documents at level A2, 553 at level B1, and 127 at level B2. As with all other corpora, back matter was removed from each text.

6.3.7 Summary and the Combined corpus (Comb.)

The distribution of documents per level are given in Table 6.4. The Combined distribution reflects a micro-trend of having the most documents at level B1 or B2, the Red Kalinka corpus being the only exception to the trend. The LingQ corpus is by far the largest at every level, with only Zlatoust level B1 on the same order of magnitude. Excluding the LingQ corpus, level C1 and C2 have the fewest document, with only 34 and 61, respectively.

Table 6.5 shows the average document length (in words) per level in each of the corpora. The overall average document size is 916 words, with a standard

	Total	A1	A2	B1	B2	C1	C2
CIE	145	28	57	60	–	–	–
news	50	–	–	–	–	–	50
LingQ	3481	323	653	716	832	609	348
RK	99	40	18	17	18	6	–
TORFL	168	31	36	36	26	28	11
Zlat.	746	–	66	553	127	–	–
Comb.	4689	422	830	1382	1003	643	409

Table 6.4: Distribution of documents per level for each corpus

deviation of 1740. Within each corpus, average document length tends to increase with each level. The CIE, TORFL and Zlatoust corpora are the clearest examples of this trend, whereas the other corpora have one or two departures from the trend.

	Total	A1	A2	B1	B2	C1	C2
CIE	362	175	369	444	–	–	–
news	186	–	–	–	–	–	186
LingQ	1116	77	78	299	1293	3729	711
RK	263	161	297	441	278	294	–
TORFL	242	87	220	249	281	297	493
Zlat	381	–	203	372	516	–	–
Comb.	916	92	119	335	1150	3548	641

Table 6.5: Average words per document for each level of each corpus

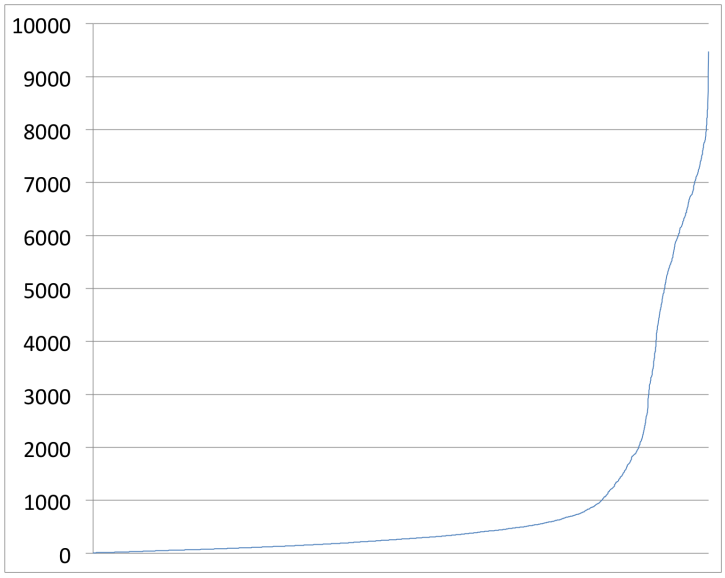
The overall distribution of document length is shown in Figure 6.1, where the x-axis is all documents ranked by document length and the y-axis is document length. The shortest document contains 7 words, and the longest document contains over 9000 words. Of the 857 documents with word length over the average of 916, 820 are in LingQ, 27 are in the Zlatoust corpus, 7 are in TORFL, and 3 are in the CIE corpus.

6.4 Features

In the following sections, I give an overview of the features used in this study, both the rationale for their inclusion, as well as details regarding their operationalization and implementation. I combine features used in previous research with some novel features based on morphological analysis.

One of the primary research questions of this study is to determine what kind

Figure 6.1: Distribution of document length in words



of information is most informative for the automatic readability classification task for Russian. This question can be asked in two different ways: 1) What kinds of natural language processing technology yield the most useful information?; 2) What level of linguistic complexity is most predictive of second language readability (lexicon, morphology, syntax, etc.)? The first question yields clear categories based on whether frequency lists, vocabulary lists, sentence boundary detection, lemmatization, part-of-speech tagging, morphological analysis, or syntactic parsing. However, from a linguistic point of view, these factors are not necessarily very interesting. On the other hand, the second question is intended to yield an answer that is relevant to linguistic theory of language complexity, but determining whether a given feature belongs to the lexicon, morphology, or syntax is not always straightforward. For example, many of the features that have traditionally been used in readability analysis are normalized by sentences. For instance, the average number of syllables per sentence is sometimes used as a measure of lexical difficulty, or short-term memory chunking capacity. However, since the sentence is a syntactic construct, such features at least partially reflect syntactic complexity. Because such features can exhibit a mixture of possible sources of processing complexity, categorizing the features according to theoretical linguistic categories is somewhat problematic.

However, in general, each feature is primarily intended to indicate one main source of complexity. Even if average syllables per sentence can potentially reflect

a degree of syntactic complexity, it is hardly a reliable indicator of syntactic complexity. For the purposes of this study, I assume that each feature has one primary linguistic domain that it is reflective of, and I categorize it accordingly. It should be remembered, however, that these categories are not as discrete as they may appear.

I divide features into the following categories: lexical, morphological, syntactic, and semantic.

6.4.1 Lexical features (LEX)

The lexical features (LEX) are divided into three subcategories: lexical variability (LEXV), lexical complexity (LEXC), and lexical familiarity (LEXF).

Lexical variability (LEXV)

As the name suggests, the lexical variability category (LEXV) contains features that are intended to measure the variety of lexemes found in a document. One of the most basic measures of lexical variability is the type-token ratio, which is the number of unique wordforms divided by the number of tokens in a text. However, because the simple type-token ratio is dependent on document length, several metrics have been proposed as more robust variations of the type-token ratio. For example, Vajjala and Meurers (2012) applied the metrics of Lu (2012, and references therein) to the readability classification task. I use these same measures here, which include the simple TTR (T/N), Root TTR ($T/\sqrt{N}$), Corrected TTR ($T/\sqrt{2N}$), Bilogarithmic TTR ($\log T/\log N$), and the Uber Index ($\log^2 T/\log(N/T)$). For all of these metrics, a higher score signifies higher concentrations of unique tokens, which indicates more difficult readability levels.

Abbr.	Formula	Explanation
TTR	T/N	Type-token ratio
RTTR	$T/\sqrt{N}$	Root type-token ratio
CTTR	$T/\sqrt{2N}$	Corrected type-token ratio
BTTR	$\log T/\log N$	Bilogarithmic type-token ratio
UberInd	$\log^2 T/\log(N/T)$	Uber index
T_{lem}TR	T_{lem}/N	Lemma type-token ratio
CT_{lem}TR	T_{lem}/N	Content lemma type-token ratio

Table 6.6: Lexical variability features (LEXV)

Lexical complexity (LEXC)

Lexical complexity includes multiple concepts. One is the degree to which individual words can be parsed into component morphemes. This is a reflection of the derivational or agglutinative structure of words. Another measure of lexical complexity is word length, which reflects the difficulty of chunking and storing words in short-term memory. Depending on the particulars of a given language or the development level of a given learner, lexical complexity can either inhibit or enhance comprehension. For example, the word *neftepererabatyvajuščij (zavod)* 'oil-refining (factory)' is overwhelming for a beginning learner, but an advanced learner who has never seen this word can easily deduce its meaning by recognizing its component morphemes: *nefte-pere-rabat-yva-juščij* 'oil-re-work-IMPF-ing'.

I computed the following features to capture variation in lexical complexity. Regarding word length, features were computed for characters, syllables, and morphemes. For each of these three, both an average and a maximum were computed. In addition, all six of these features were computed for both all words, and for content words only.[8] Russian orthography is such that almost all vowels are syllabic, so syllable counts are, in fact, vowel counts. Exceptions to this generalization are rare enough that this simplification should be quite reliable. The features for word length in morphemes were computed on the basis of Tixonov's Morpho-orthographic dictionary (Tixonov, 2002), which contains parses for about 100 000 words. All words that are not found in the dictionary were ignored.

In addition to average and maximum word lengths, I also followed Karpov et al. (2014) in calculating word length bands, such as the proportion of words with five or more characters. These bands are calculated for 5–13 characters (9 features) and 3–6 syllables (4 features). All 13 of these features were calculated both for all words and for content words only.

Lexical familiarity (LEXF)

A number of features were computed to attempt to capture the degree to which the words of a text are familiar to readers of various levels. These features model the development of learners' vocabulary from level to level. Unlike the features for lexical variability and lexical complexity, which are primarily based on surface structure, the features for lexical familiarity are primarily based on abstracted facts about given surface forms, typically by referencing a frequency list or lexicon of some kind.

The first set of lexical familiarity features are derived from the official "Lexical

[8]The following parts of speech were considered content words: adjectives, adverbs, nouns and verbs.

Abbr.	Formula	Explanation
WL_{char}	char/N	Average word length (characters)
CWL_{char}	$\text{char}_{cont}/N_{cont}$	Average content word length (characters)
$\text{maxWL}_{\text{char}}$	$\max(\text{char})/N$	Maximum word length (characters)
$\text{CmaxWL}_{\text{char}}$	$\max(\text{char})_{cont}/N_{cont}$	Maximum content word length (characters)
WL_{syll}	vowel/N	Average word length (syllables)
CWL_{syll}	$\text{vowel}_{cont}/N_{cont}$	Average content word length (syllables)
$\text{maxWL}_{\text{syll}}$	$\max(\text{vowel})/N$	Maximum word length (syllables)
$\text{CmaxWL}_{\text{syll}}$	$\max(\text{vowel})_{cont}/N_{cont}$	Maximum content word length (syllables)
WL_{morph}	morph/N	Average word length (morphemes)
$\text{maxWL}_{\text{morph}}$	$\max(\text{morph})/N$	Maximum word length (morphemes)
$\text{Num}n_{\text{char}}$	$N_{\text{char}\geq n}/N$	Words with n or more characters ($5 \geq n \geq 13$)
$\text{CNum}n_{\text{char}}$	$N_{char\geq n}/N$	Content words with n or more characters ($5 \geq n \geq 13$)
$\text{Num}n_{\text{syll}}$	$N_{\text{syll}\geq n}/N$	Words with n or more syllables ($3 \geq n \geq 6$)
$\text{CNum}n_{\text{syll}}$	$N_{syll\geq n}/N$	Content words with n or more syllables ($3 \geq n \geq 6$)

Table 6.7: Lexical complexity features (LEXC)

Minimum" lists published by Zlatoust for learners preparing for the TORFL examinations. The lexical minimum lists are compiled for the four lowest levels (A1, A2, B1, and B2), where each list contains the words that should be mastered for the tests at each level. These lists can be seen as *prescriptive* vocabulary for language learners. Following Karpov et al. (2014), I computed features for the proportion of words above a given reading level.[9]

The second set of lexical familiarity features are taken from the Kelly Project (Kilgarriff et al., 2014), which is a "corpus-based vocabulary list" for language learners. Whereas the prescriptive lexical minimum lists define what a language learner *should* be familiar with, the Kelly Project is a *descriptive* approach to the question, compiling lists for all six CEFR levels based primarily on word frequency. The complete methodology for deriving these lists is described in the article cited above. Just like the features based on the lexical minimum, I computed the proportion of words over each of the six CEFR levels.

The third set of lexical familiarity features are based on raw frequency lists, both lemma frequency and token frequency. Lemma frequency data were taken from Ljaševskaja and Šarov (2009) (available digitally at `http://dict.ruslang.ru/freq.php`), which is based on data from the Russian National Corpus. The token frequency data were taken directly from the Russian National Corpus webpage at `http://ruscorpora.ru/corpora-freq.html`. For both kinds of frequency, I used both raw frequency and frequency rank data.[10] For each of the four kinds of frequency data, I computed average, median, minimum, and standard deviation. Once again, all of these features were calculated for all words and for content words only.

6.4.2 Morphological features (MORPH)

Morphological features are primarily based on morphosyntactic values, as output by our morphological analyzer. These features are given in Table 6.9. The first three sets of features reflect simple counts of whether a morphosyntactic tag is present or what proportion of tokens receive each morphosyntactic tag. The first set of features expresses whether a given morphosyntactic tag is present in the document. A second set of features, expresses the ratio of tokens with each morphosyntactic tag, normalized by token count. A third set of features, the value-feature ratio (VFR), was calculated as the number of tokens that express a mor-

[9] None of the documents in Karpov et al. were at the B2 level, so they used only the three lowest lexical minimum lists. Because my data span all six reading levels, I use all four lists.

[10] The frequency rank data were such that items with the same frequency were ranked differently merely because of alphabetic sorting. To avoid this bias, I also computed features based on ranking in which all items of the same frequency share the same rank.

Abbr.	Formula	Explanation
Over$X$_lm	$N_{lev \geq X}/N$	Words over *lexical minimum* level X (A1$\geq$ X $\geq$B2)
Over$X$_kp	$N_{lev \geq X}/N$	Words over Kelly Project level X (A1$\geq$ X $\geq$C2)
LF_{mean}		Mean lemma frequency
LF_{med}		Median lemma frequency
LF_{min}		Minimum lemma frequency
LF_{stddev}		Std deviation lemma frequency
LFR_{mean}		Mean lemma frequency rank
CLF_{mean}		Mean content lemma frequency
CLF_{med}		Median content lemma frequency
CLF_{min}		Minimum content lemma frequency
CLF_{stddev}		Std deviation content lemma frequency
$CLFR_{mean}$		Average content lemma frequency rank

Table 6.8: Lexical familiarity features (LEXF)

phosyntactic value (e.g. past), normalized by the number of tokens that express the corresponding morphosyntactic feature (e.g. tense).

In the early stages of learning Russian, learners do not have a knowledge of all six cases, so I hypothesized that texts at the lowest reading levels may be distinguished by a limited number of cases. Therefore, the case-coverage feature (CC) expresses the number of cases found in a document. Similarly, two subcases in Russian, partitive genitive and locative, are generally rare, but are highly overrepresented in texts written for beginners who are being introduced to these subcases. Therefore, the subcase-coverage feature (SCC) gives the number of subcases found in the document.

Following Nikin et al. (2007); Krioni et al. (2008); Filippova (2010), I calculated a feature to measure the proportion of abstract words. This was done by using a regular expression to test lemmas for the presence of a number of abstract derivational suffixes. This feature is normalized to the number of tokens in the document.

Sentence length-based features (SENT)

The SENT category consists of features that include in their computation some form of sentence length, including the traditional readability formulas. Even though sentence length is technically a syntactic feature, the following features were not included in the SYNT category for a number of reasons. From a linguistic point

Abbr.	Formula	Explanation
XPres	boolean	MS tag X present
VTRX	N_X/N	Value-token ratio (e.g., N_{nom}/N)
VFRX	N_X/N_{catX}	Value-feature ratio (e.g., N_{nom}/N_{CASE})
CC		Case coverage (values: 0-6)
SCC		Sub-case coverage (values: 0-3)
Abstr	$N_{abstract}/N$	Proportion abstract words

Table 6.9: Morphological features (MORPH)

of view, sentence length is only a superficial indication of syntactic complexity. The two are usually correlated, to be sure, but sentences of equal length can be dramatically differ in syntactic complexity.

Keeping SENT and SYNT categories separate has both linguistic and technical benefits, since it also makes a distinction between features that can be extracted with or without different kinds of natural language processing technology. The features in SENT can be extracted using sentence boundary detection and morphological analysis. The features in the SYNT category are extracted using the output of a syntactic dependency parser.

The SENT features include words per sentence, syllables per sentence, letters per sentence, coordinating conjunctions per sentence, and subordinating conjunctions per sentence. In addition, I also compute the number of unique morphosyntactic readings per sentence, which I call the reading type frequency. Finally, the SENT category also includes the traditional readability formulas: Russian Flesch Reading Ease (Oborneva, 2006a), Flesch Reading Ease, Flesch-Kincaid Grade Level, and the Coleman-Liau Index. All these features are summarized in Table 6.10.

6.4.3 Syntactic features (SYNT)

Syntactic features for this study were primarily based on the output of the `hunpos`[11] trigram part-of-speech tagger, which served as input to the `maltparser`[12] syntactic dependency parser, both trained on the SynTagRus[13] treebank. Using `maltoptimizer`,[14] I found that the best-performing algorithm was Nivre Eager, which achieved a labeled attachment score of %81.29 with cross-validation of SynTagRus. This level of accuracy is high enough to reliably serve as the basis for

[11] `https://code.google.com/p/hunpos/`
[12] `http://www.maltparser.org/`
[13] `http://ruscorpora.ru/instruction-syntax.html`
[14] `http://nil.fdi.ucm.es/maltoptimizer/index.html`

Abbr.	Formula	Explanation
SL_{word}	N/N_{sent}	Sentence length in words
SL_{syll}	N_{syll}/N_{sent}	Sentence length in syllables
SL_{char}	N_{char}/N_{sent}	Sentence length in letters
CC/S	N_{coord}/N_{sent}	№ of coord. conj. per sentence
SC/S	N_{subord}/N_{sent}	№ of subord. conj. per sentence
T_{read}/S	T_{read}/N_{sent}	Reading type frequency per sentence
FRER	$206.836 - (1.3 * \text{WL}_{\text{syll}}) - (60.1 * \text{SL}_{\text{word}})$	Flesch Reading Ease (Russ.)
FRE	$206.835 - (1.015 * \text{WL}_{\text{syll}}) - (84.6 * \text{SL}_{\text{word}})$	Flesch Reading Ease
FKGL	$-15.59 + (11.8 * \text{WL}_{\text{syll}}) + (0.39 * \text{SL}_{\text{word}})$	Flesch-Kincaid Grade Level
CLI	$-15.8 + (5.88 * \text{WL}_{\text{syll}}) - (29.6/\text{SL}_{\text{word}})$	Coleman-Liau Index

Table 6.10: Features calculated on the basis of sentence length (SENT)

feature extraction.

Researchers of automatic readability classification and closely related tasks have used a number of syntactic dependency features which I also implement here (Yannakoudakis et al., 2011; Dell'Orletta et al., 2011; Vor der Brück and Hartrumpf, 2007; Vor der Brück et al., 2008). These include features based on dependency lengths (the number of tokens intervening between a dependent and its head), as well as the number of dependents belonging to particular parts of speech, in particular nouns and verbs. In addition, I also include features based on dependency tree depth (the path length from root to leaves).

For each sentence, I recorded the average and maximum of four values: dependency length, tree depth, number of verbal dependents, and number of noun dependents. Then for each document, I compute the average and maximum of each of these properties. The average is an arithmetic average of every instance of the property in the document. The maximum, on the other hand, represents only one instance—usually only part of a sentence—that is the most complex in the document. These two metrics have inherent weaknesses. The arithmetic average has a tendency to wash out differences between datasets, and the maximum is frequently expected to be an outlier—and therefore not representative of the document as a whole. Therefore, in addition to the average and maximum, I compute two additional features for each property: the maximum of the sentence-level averages, and the average of the sentence-level maximums. The maximum of averages tells us the level of complexity of the sentence which is on average the most complex in the document. The average of maximums gives us the average complexity of only the most complex part of each sentence.

Abbr.	Explanation
DLavg	Average of each sentence's average dependency length
DLmax	Maximum dependency length
max(DLavg)	Maximum of all sentences' average dependency lengths
avg(DLmax)	Average of all sentences' maximum dependency lengths
TDavg	Average of each sentence's average tree depth
TDmax	Maximum dependency length
max(TDavg)	Maximum of all sentences' average tree depth
avg(TDmax)	Average of all sentences' maximum tree depths
VDavg	Average of all sentences' average number of verb dependents
VDmax	Maximum number of verb dependents
max(VDavg)	Maximum of all sentences' average number of verb dependents
avg(VDmax)	Average of all sentences' maximum number of verb dependents
NDavg	Average of all sentences' average number of noun dependents
NDmax	Maximum number of noun dependents
max(NDavg)	Maximum of all sentences' average number of noun dependents
avg(NDmax)	Average of all sentences' maximum number of noun dependents

Table 6.11: Syntactic features (SYNT)

6.4.4 Discourse/content features (DISC)

The final category of features is the discourse/content features (DISC). These features are intended to capture the broader difficulty of understanding the text as a whole, rather than the difficulty of processing the linguistic structure of particular words or sentences. One set of features are based on *definitions*, an idea taken directly from Krioni et al. (2008), which defines a number of words and phrases[15] that are used to introduce or define new terms in a text. As features, I include definitions per token and definitions per sentence.

Another set of features is adapted from the work of Brown et al. (2007, 2008), who show that propositional density—a fundamental measurement in the study of discourse comprehension—can be accurately measured purely on the basis of English part-of-speech counts. More specifically, adjectives, adverbs, prepositions, conjunctions, determiners (except articles), modals (if negated), and verbs (except auxiliary and linking verbs) all count as logical propositions. Although the psychological research on propositional density is primarily supported by English data, it

[15] Krioni et al. list the following constructions: "____ – *eto*", "____ *est'* ____", "____ – ____", "*nazyvaetsja*", "*nazyvajutsja*", "*ponimaetsja*", "*ponimajutsja*", "*predstavljaet soboj*", "*predstavljajut soboj*", "*o(bo)značaet*", "*o(bo)značajut*", "*opredeljaet(sja)*", "*opredeljajut(sja)*", "*sčitaet(sja)*", and "*sčitajut(sja)*".

is reasonable to assume that a similar approach in Russian can yield a meaningful measure of discursive complexity. Brown et al. propose a number of adjustments based on analytical syntactic constructions in English grammar that do not exist in Russian, so my implementation of the approach with Russian is even simpler than their original proposal, based purely on part-of-speech counts, with no exceptions.

One other feature is based on the intuition that reading dialogic texts is generally easier than reading prose. This feature is computed as the number of dialog symbols[16] per token.

Abbr.	Formula	Explanation
DTR	N_{def}/N	Definitions per token
D/S	N_{def}/N_{sent}	Definitions per sentence
PD_{tok}		Proposition density per token
PD_{sent}		Proposition density per sentence
Dial		Dialog punctuations per token

Table 6.12: Discourse features (DISC)

6.4.5 Summary of features

As outlined in the preceding sections, this study makes use of 179 features. Many of the features are inspired by previous research of readability, both for Russian and for other languages. The distribution of these features across categories is shown in Table 6.13.

Category	Number of features
DISC	6
LEXC	42
LEXF	38
LEXV	7
MORPH	60
SENT	10
SYNT	16
Total	179

Table 6.13: Distribution of features across categories

[16] In Russian, -, –, and : are used to mark turns in a dialog.

6.5 Results

The machine-learning and evaluation for this study were performed using the `weka` data mining software. For the sake of consistency and comparability, I wanted to select one classifier algorithm to use throughout the study. I performed ten-fold cross-validation using a variety of models and using different combinations of subcorpora. The Random Forest model was consistently among the highest-performing models, and so it was selected as the classifier algorithm for the study.[17] All results reported below are achieved using the Random Forest algorithm with default parameters. Unless otherwise specified, evaluation was performed using ten-fold crossvalidation.

The basic results are given in Table 6.14. *Precision* is a measure of how many of the documents predicted to be at a given readability level are actually at that level, mathematically expressed as true positives divided by true and false positives. *Recall* measures how many of the documents at a given readability level are predicted correctly, mathematically expressed as true positives divided by true positives and false negatives. The two metrics are calculated for each reading level and a weighted average is reported for the classifier as a whole. The *F-score* is a harmonic mean of precision and recall. *Adjacent accuracy* is the same as weighted recall, except that it considers predictions that are off by one category as correct. For example, a B2 document is counted as being correctly classified if the classifier predicts B1, B2, or C1. The baseline performance achieved by predicting the mode reading level (B1)—using `weka`'s ZeroR classifier—is precision 0.097 and recall 0.312 (F-score 0.149). The OneR classifier, which is based on only the most informative feature (corrected type-token ratio), achieves precision 0.487 and recall 0.497 (F-score 0.471). The Random Forest classifier, trained on the full Combined corpus with all 179 features, achieves precision 0.69 and recall 0.677 (F-score 0.671) on ten-fold cross-validation.

Classifier	Precision	Recall	F-score
ZeroR	0.097	0.312	0.149
OneR	0.487	0.497	0.471
RandomForest	0.690	0.677	0.671

Table 6.14: Baseline and RandomForest results with Combined corpus

An F-score of 0.671 is a modest result, and in order to see where the classifier is going wrong, I give a confusion matrix in Table 6.15, which shows the predic-

[17] Other classifiers that consistently performed well were NNge (nearest-neighbor with non-nested generalized exemplars), FT (Functional Trees), MultilayerPerceptron, and SMO (sequential minimal optimization for support vector machine).

tions of the RandomForest classifier. The rows represent the actual reading level as specified in the gold standard, whereas the columns represent the reading level predicted by the classifier. Correct classifications appear along the diagonal. Table 6.15 shows that the majority of misclassifications are only off by one level, and indeed the adjacent accuracy is 0.919, which means that less than 10% of the documents are more than one level away from the gold standard. When one considers the hetergeneous nature of the corpus, coming from many different sources, and representing many different genres, this result is actually suprisingly high. In the following section, I take a closer look at the quality of the corpus as a whole, as well as individual subcorpora.

	A1	A2	B1	B2	C1	C2	<– classified as
A1	234	120	48	0	0	0	
A2	41	553	192	17	0	0	
B1	16	76	1130	90	5	5	
B2	1	57	311	478	83	4	
C1	1	20	66	98	394	6	
C2	0	3	40	58	9	78	

Table 6.15: Confusion matrix for RandomForest, all features, Combined corpus

As shown in Section 6.3.7, the documents in the Combined corpus vary significantly with regard to document length and document distribution across reading levels. It is well known that features such as type-token ratio are significantly affected by document length, and as I will discuss later, the type-token ratio features are among the most informative of my feature set. This means that a skew in type-token ratios could significantly change classification outcomes.

In order to control whether document length adversely affects classification accuracy, I truncated each document to the sentence boundary nearest the 300-word mark, and recalculated each document's features. The 300-word limit leaves a proportion of documents with fewer than 300 words, so the corpus is not truly leveled. However, since there are documents that consist of fewer than 10 words, leveling the entire corpus to such a small size would certainly throw away too much information to expect any gains. This resulted in precision 0.635 and recall 0.624 (F-score 0.615), which is almost 6% lower than using full documents.

The reading level with the fewest documents is C2, with 409 documents. Therefore, to control for document distribution, I computed features for a corpus that included only 409 randomly selected documents at each level. This also resulted in slightly less accurate predictions: precision 0.675 and recall 0.662 (F-score 0.655).

These results indicate that removing data in order to balance document length or distribution will not lead to improved predictions. Therefore, for the remainder

of the study, I do not attempt to balance the corpus using either of these methods.

6.5.1 Corpus evaluation

This study introduces several new readability gold-standard corpora, and before one can begin to evaluate the effectiveness of readability classification models in much detail, it is important to establish the quality of the gold-standard corpora themselves, including the inter-corpus validity of reading levels. In this section, I explore both the internal consistency of each individual corpus, as well as the accuracy with which a model trained on one subcorpus can predict the readability of other subcorpora.

To begin, we look at a train-test matrix containing F-scores for each of the subcorpora, given in Table 6.16.[18] Rows show the training corpus and columns show the test corpus. Along the diagonal—where the training and test corpus is the same—ten-fold cross validation is used. It is important to note that in all other cases where an apparent overlap exists between training and test corpus, the larger of the two corpora has the overlapping instances removed. For example, when the training corpus is LingQ, and the test corpus is the Combined corpus (which contains LingQ), the classifier is trained on LingQ and tested on the Combined corpus with LingQ removed.

Two corpora are grayed out—CIE and Zlat.—because they contain documents at only three levels, so the probability of correct classifications is much higher. This makes the results with these two corpora less comparable with the results of the remaining corpora. The Red Kalinka is not grayed out, even though it does not have documents at the C2 level, so its comparability is also dimished, although to a lesser degree.

The results along the diagonal in Table 6.16 show that in general, each subcorpus achieves a higher F-score in ten-fold cross-validation than the Combined corpus. I interpret this result to mean that for the most part, each subcorpus is more consistent internally than the Combined corpus, which should be expected. This trend has two surprising exceptions. The CIE corpus achieves only 0.608 and the TORFL corpus only 0.501.

The fact that the CIE corpus achieves such a low F-score is surprising because it contains documents at only three different levels, so we should expect its results to be higher than the other subcorpora. Assuming that the features used in this study capture the essential properties of readability, this indicates that the ratings in the CIE corpus are internally inconsistent.

[18]The news corpus was included in the Comb. training and test corpora, but it could not be considered independently because all of its documents are on the C2 level.

	Comb.	LingQ	RK	CIE	TORFL	Zlat.	Evgenij	LQsupp	Expert	lq
Comb.	**0.671**	0.193	0.426	0.383	0.204	0.649	0.310	0.313	0.203	0.178
LingQ	0.487	**0.682**	0.411	0.415	0.254	0.653	0.234	0.321	0.222	0.184
RK	0.177	0.134	**0.849**	0.397	0.270	0.528	0.190	0.139	0.094	0.151
CIE	0.452	0.502	0.574	**0.608**	0.643	0.219	0.530	0.533	0.430	0.492
TORFL	0.226	0.230	0.417	0.594	**0.501**	0.208	0.296	0.091	0.288	0.096
Zlat.	0.279	0.277	0.228	0.000	0.343	**0.798**	0.328	0.280	0.187	0.386
Evgenij	0.355	0.289	0.171	0.371	0.216	0.689	**0.735**	0.401	0.228	0.174
LQsupp	0.243	0.258	0.402	0.348	0.236	0.498	0.272	**0.794**	0.248	0.164
Expert	0.253	0.245	0.405	0.329	0.207	0.245	0.219	0.333	**0.788**	0.213
lq	0.304	0.275	0.156	0.299	0.084	0.639	0.226	0.361	0.311	**0.768**

Table 6.16: Train-test matrix for all subcorpora, showing F-scores from RandomForest with all features

The fact that the TORFL corpus achieves such a low F-score is especially surprising because its documents come from the most authoritative source of any of the corpora: the official TORFL proficiency tests. This seems to be the result of the conspicuous fact that these texts come from tests, which means that the test writers can easily adjust the difficulty of the task to adjust for an overly difficult text. The readability of a document is directly related to what the reader is expected to do with that document. In the general case, the reader is expected to read and understand the text as a whole. However, in the case of the TORFL corpus, each text is accompanied by tasks that determine which parts of the text a learner must understand, which can lower the required reading aptitude.

As for the cross-validation performance of other subcorpora, the Red Kalinka corpus achieves the highest F-score (0.849), with an adjacent accuracy of 0.98. The LingQ corpus achieves a slightly higher F-score than the Combined corpus, but each of its subcorpora (Evgenij, LQsupp, Expert, and lq) individually achieves significant gains over the Combined and LingQ corpora.

Perhaps the most striking pattern in Table 6.16 is the sharp drop in F-scores between intra-corpus cross-validation and inter-corpus training/testing. Whereas most of the cross-validation cells along the diagonal are near or above 0.700, very few of the cells that train on one corpus and test on another score higher than 0.400. Indeed, many of these subcorpus combinations' F-scores are at or near the ZeroR baseline.

The fact that the trained models do not perform well on external texts could be indicative of many things. It may be that overfitting is responsible for the drop in performance. If the model is too specific, then it will not generalize well to new data. However, overfitting is not a likely cause here, since the Random Forest

algorithm was specifically designed to overcome the tendency of Random Trees to overfit (Breiman, 2001). Other possible causes include the possibility that either the validity of classes between subcorpora is low or misaligned, or the features that are informative for one dataset are not informative for other datasets. The information carried by each feature will be explored in more detail below, but first we take up the question of inter-corpus validity.

Validity of ratings between corpora

There is little guarantee that the humans assigning reading levels to each of these documents did so according to externally verifiable criteria. For those documents that come from institutions—such as Red Kalinka, Zlatoust, or LQsupp—it is reasonable to expect that each document's readability was determined in some methodical way. However, even if these ratings were assigned in a methodical way, the alignment of readability levels between subcorpora is far from certain. Based on the same results from which Table 6.16 was derived, I also computed a Spearman rank correlation coefficient, which assesses how well the relationship between the actual level and predicted level can be described using a monotonic function. A high correlation means that both sets preserve the same order of elements when they are sorted according to reading level. For example, a classifier that consistently predicts one level too high would have a perfect Spearman correlation of 1.0, even though its accuracy is 0.0. Likewise, a classifier that always predicts one level too low, would have a perfect negative Spearman correlation of -1.0, even though its accuracy is 0.0. The results are given in Table 6.17.

	Comb.	LingQ	RK	CIE	TORFL	Zlatoust	Evgenij	LQsupp	Expert	lq
Comb.	**0.796**	0.559	0.737	0.549	0.432	0.307	0.680	0.398	0.653	0.196
LingQ	0.390	**0.834**	0.661	0.535	0.427	0.289	0.631	0.419	0.632	0.239
RK	0.458	0.574	**0.932**	0.504	0.384	0.129	0.604	0.565	0.415	0.583
CIE	0.487	0.505	0.577	**0.585**	0.647	0.107	0.560	0.501	0.270	0.535
TORFL	0.513	0.507	0.497	0.542	**0.610**	0.158	0.496	0.258	0.385	0.250
Zlatoust	.173	.196	.240	nan	.320	**0.596**	.119	.358	-0.086	.423
Evgenij	0.589	0.588	0.524	0.533	0.460	0.262	**0.844**	0.431	0.691	0.264
LQsupp	0.524	0.649	0.635	0.458	0.435	0.249	0.699	**0.805**	0.445	0.267
Expert	0.536	0.638	0.628	0.071	0.369	0.108	0.558	0.446	**0.833**	0.106
lq	0.316	0.295	0.420	0.278	0.287	0.263	0.428	0.501	0.044	**0.741**

Table 6.17: Train-test matrix for all subcorpora, showing Spearman's Rho from RandomForest with all features

As expected, the Red Kalinka corpus has the highest rank correlation on cross-validation (0.932). The two lowest rank correlations on cross-validation are CIE

and Zlatoust, both of which have only three levels. It may be that with only three possible ranks, establishing correlation is more difficult. Also, as expected, the TORFL corpus has lower correlation on cross-validation. Otherwise, the remaining corpora have correlations approaching or above 0.800, which indicates a reasonable level of internal consistency.

As for inter-corpus correlation, overall, Zlatoust has the worst correlation with other corpora, with an average 0.26 correlation as predictor, and 0.247 correlation when predicted by models trained on other corpora. The lq corpus also has lower correlations, with an average of 0.36 correlation, both as predictor and predicted. For all other subcorpora, the average of their correlations with other subcorpora ranges from 0.400 to 0.547.[19]

One other way to compare reading level alignment between subcorpora is to compute the difference between the average of the predicted classes and the average actual class. In other words, on average, does a given model predict higher or lower than the actual reading level? These data are shown in Table 6.18. Positive values indicate that on average the model predicts higher than the actual rating, and negative values lower.

	Comb.	LingQ	RK	CIE	TORFL	Zlatoust	Evgenij	LQsupp	Expert	lq
Comb.	**-0.112**	-0.730	0.071	0.586	0.030	-0.142	0.078	-0.261	-0.639	-1.207
LingQ	-0.078	**-0.107**	0.000	0.517	-0.274	0.046	0.333	-0.160	-0.569	-1.042
RK	-0.794	-0.991	**-0.071**	0.145	-0.484	0.279	-0.669	-1.038	-1.409	-1.107
CIE	-0.369	-0.267	-0.080	**0.014**	0.262	-0.656	-0.359	-0.156	0.023	-0.374
TORFL	-0.871	-0.861	-0.566	-0.255	**-0.274**	-0.881	-0.719	-0.908	-1.025	-1.980
Zlatoust	-0.069	-0.129	-0.038	0.487	0.235	**-0.087**	0.192	-0.569	-0.501	-0.327
Evgenij	-0.391	-0.588	0.303	0.559	-0.375	-0.092	**-0.085**	-0.484	-0.651	-1.323
LQsupp	0.154	0.011	0.010	0.524	-0.048	0.283	0.228	**-0.100**	-0.284	-0.802
Expert	0.352	0.316	0.747	0.455	0.012	0.736	0.433	0.053	**0.007**	-0.890
lq	1.132	1.099	0.545	0.697	1.810	0.013	1.678	0.733	0.535	**0.268**

Table 6.18: Train-test matrix for all subcorpora, showing difference between predicted and actual average reading level from RandomForest with all features

The cells along the diagonal in Table 6.18 show that small differences between predicted and actual averages are possible even when training and testing on the same corpus. For example, on cross-validation TORFL predicts -0.274 and lq pre-

[19]This would seem to indicate that removing Zlatoust and/or lq from the Combined corpus would result in a higher F-score. Removing lq from the Combined corpus yields only a very slight improvement to the F-score (0.673), which is most likely statistically equivalent with the full Combined corpus (0.671). The other options—removing Zlatoust or removing both Zlatoust and lq—both result in lower F-scores.

dicts +0.268. Such small differences are not necessarily meaningful, but there are some trends in this table that are noteworthy. The models trained on the lq corpus predict, on average, almost an entire step higher (0.851) than the actual reading levels. At the same time, models trained on other subcorpora predict lq data, on average, almost an entire step lower (-0.8784). This is strong evidence that lq's reading levels do not align with other subcorpora. The Expert corpus has a similar trend, but to a lesser degree in both directions (0.2221 and -0.4513). The CIE corpus exhibits the trend of the same magnitude as Expert, but in the opposite direction (-0.1962 and 0.3729). Models trained on the Red Kalinka and TORFL subcorpora predict other subcorpora low (-0.6139 and -0.834, respectively).

Summary of corpus evaluation The main purpose of evaluating the differences between subcorpora is to establish stability for further evaluation of learning curves and features. For instance, the distribution of documents from various subcorpora across levels in the Combined corpus is not consistent; at any given reading level some subcorpora are overrepresented, while other are lacking altogether. When evaluating the feature set at different levels, it is impossible, then, to determine whether the observed effects are an outcome of level variation or subcorpus variation. In order to minimize the conflation of variables, it is desirable to limit the data to the subcorpus or subcorpora that are the most consistent and valid.

According to almost every metric discussed above, the Red Kalinka corpus gets the best results. It gets the highest F-score on cross-validation, and it has the highest rank correlation with itself on cross-validation. It has the highest average rank correlation when predicted by models trained on other corpora, and its average rank correlation as training corpus is less than 0.020 below the highest. Unfortunately, the Red Kalinka corpus does not have any documents at the C2 level, which may itself be a partial explanation for why its metrics are consistently higher than those of the other subcorpora.

Determining the next best subcorpus is not as straightforward, but according to the metric presented above, both Evgenij and LQsupp are relatively consistent and valid, and they come from sources that are reputable. The author of the Evgenij corpus is a professional Russian teacher and LQsupp corpus comes for the official contributor to Russian on the LingQ website. Based on my own subjective impressions of each subcorpus during data cleaning and preprocessing, I believe that the Evgenij corpus contains many texts that are not prototypical of the kinds of authentic texts that would be used in an ATICALL environment. Therefore, for evaluation of the feature set used in this study, I will use the LQsupp subcorpus as the gold-standard corpus.

6.5.2 Binary classifiers

Evaluation was performed with binary classifiers, in which the datasets contain only two adjacent readability levels. Since the Combined corpus has six levels, there are five binary classifier pairs: A1-A2, A2-B1, B1-B2, B2-C1, C1-C2. The results of the cross-validation evalution of these classifiers is given in Table 6.19.

		A1-A2	A2-B1	B1-B2	B2-C1	C1-C2
Combined	precision	0.821	0.857	0.817	0.833	0.894
	recall	0.821	0.857	0.811	0.831	0.897
	F-score	0.812	0.855	0.806	0.826	0.892
Red Kalinka	precision	0.967	0.943	0.832	0.837	–
	recall	0.966	0.943	0.829	0.792	–
	F-score	0.965	0.943	0.828	0.730	–
LQsupp	precision	0.911	0.806	0.955	0.914	0.926
	recall	0.903	0.806	0.956	0.915	0.924
	F-score	0.901	0.806	0.954	0.912	0.924

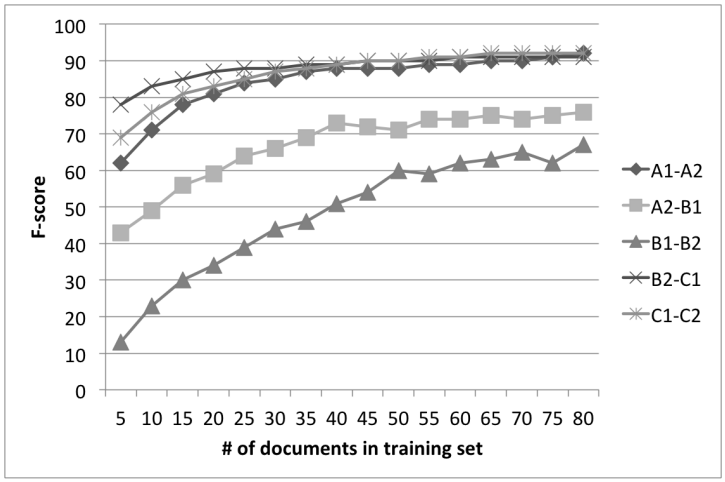
Table 6.19: Evalution metrics for binary classifiers: RandomForest, Combined corpus, all features

As expected, because the binary classifiers' are more specialized, with less data noise and fewer levels to choose between, their accuracy is much higher.

One potentially interesting difference between binary classifiers at different levels is their learning curves, or in other words, the amount of training data needed to achieve similar results. I hypothesize that the binary classifiers at lower levels need less data, because texts for beginners have limited possibilities for how they can vary without increasing complexity. Texts at higher reading levels, however, can vary in many different ways. To adapt Tolstoy's famous opening line to *Anna Karenina*, "All [simple texts] are similar to each other, but each [complex text] is [complex] in its own way." If this is true, then binary classifiers at higher reading levels should require more data to reach the upper limit of their classifying accuracy. This prediction was tested by controlling the number of documents used in the training data for each binary classifier, while tracking the F-score on cross-validation. Results of this experiment are given in Figure 6.2.

The results of this experiment support the hypothesized difference between binary classifier levels, albeit with some exceptions. The A1-A2 classifier rises quickly, and begins to level off after seeing about 40 documents. The A2-B1 classifier rises more gradually, and levels off after seeing about 55 documents. The B1-B2 classifier rises even more slowly, and does not level off within the scope of this line chart.

Figure 6.2: Learning curves of binary classifiers trained on LQsupp subcorpus



Up to this point, the data confirm my hypothesis that lower levels require less training data. However, the final two binary classifiers buck this trend, with learning curves that outperform the simplest binary classifier with very little training data. One possible explanation for this is that the increasing complexity of CEFR levels is not linear, meaning that the leap from A1 to A2 is much smaller than the leap from C1 to C2. In at least one aspect, the increasing rate of change is explicitly formalized in the official standards for the TORFL tests. For example, the number of words that a learner should know has the following progression: 750, 1300, 2300, 10 000, 12 000 (7 000 active), 20 000 (8 000 active). This means that distinguishing B2-C1 and C1-C2 should be easier because the distance between their respective levels is an order of magnitude larger than the distance between the respective levels of A1-A2, A2-B1. Furthermore, development of grammar should be more or less complete by level B2, so that the the number of features that distinguish C1 from C2 should be smaller than in lower levels, where grammar development is a limiting factor.

In order to test this hypothesis, the following section examines the informativeness of various features for both six-level and binary classifiers.

6.6 Feature evaluation

As summarized in Section 6.4.5, this study makes use of 179 features, divided into 7 categories: DISC, LEXC, LEXF, LEXV, MORPH, SENT, and SYNT. Many of the features used in this study are taken from previous research of related topics,

and some features are proposed for the first time here. To my knowledge, previous researchers of Russian readability have not included morphological features, so the results of these features is of particular interest here.

In this section, I explore the extent to which the selected corpora can support the relevance and impact of these features in Russian second language readability classification. I also explore which features are most important at which reading levels.

One rough test for the value of each category of features is to run cross-validation with models trained on only one category of features. In Table 6.20, I report the results of this experiment using the Combined corpus.

Category	# features	precision	recall	F-score
DISC	6	0.482	0.482	0.477
LEXC	42	0.528	0.532	0.514
LEXF	38	0.581	0.573	0.567
LEXV	7	0.551	0.552	0.546
MORPH	60	0.642	0.627	0.618
SENT	10	0.478	0.479	0.474
SYNT	16	0.518	0.533	0.514
LEXC+LEXF+LEXV	87	0.652	0.645	0.639

Table 6.20: Precision, recall, and F-score for six-level Random Forest models trained on the Combined corpus

The results in Table 6.20 show that MORPH, has the highest F-score of any single category, with an F-score just 0.053 below a model trained on all 179 features. True comparisons between categories are problematic because the number of features per category varies significantly. However, even when the all 87 lexical features are included in the training data, the results are only slightly better than MORPH. This supports the conclusion that morphological factors are important for readability classification.

In order to evaluate how much information is provided by each feature, I use weka's InformationGainAttributeEval using the Ranker search method. Feature evaluation is independent of any classifier algorithm. Table 6.21 shows the top 30 features, ranked by information gain. Note that many of the features not included in this list still contribute significant information; only 32 features have less than 0.10.

The top half of Table is dominated by LEXV, SYNT, and LEXC features, which quickly yield to MORPH features in the second half of the list. One striking feature of this list is how many of the features represent some kind of maximum, especially

InfoGain	feature
0.64915	LEXV_RTTR
0.649	LEXV_CTTR
0.49677	SYNT_maxDepLen
0.4677	SYNT_maxTreeDep
0.46342	SYNT_maxavgTreeDep
0.39759	LEXV_TlemTR
0.38942	SYNT_maxavgDepLen
0.38485	TXT_dialogSymbCount
0.35759	LEXV_TTR
0.35393	LEXC_CmaxWLchar
0.35295	LEXC_maxWLchar
0.35264	LEXC_CmaxWLsyll
0.35264	LEXC_maxWLsyll
0.33922	SYNT_maxVDeps
0.33407	LEXC_maxWLmorph
0.318	MORPH_actvPerV
0.31006	MORPH_pprsPerV
0.3057	MORPH_ppPerV
0.29429	SYNT_maxNDeps
0.2843	MORPH_pasvPerV
0.27739	SYNT_avgmaxTreeDep
0.26979	LEXV_CTlemTR
0.25857	MORPH_CSPerWord
0.25513	MORPH_CSperPOS
0.25288	SYNT_avgmaxDepLen
0.24927	SYNT_maxavgVDeps
0.24638	MORPH_prctIns
0.24595	LEXV_BTTR
0.24454	SENT_CS/S
0.24416	LEXF_prctOverA1_lm

Table 6.21: Top 30 features ranked by information gain, Combined corpus, all levels

the SYNT and LEXC features. This is a little troubling because a maximum is a very volatile property of a text. A large text could have only one outlier to be classified differently. Relying too heavily on maximums could make a model more error-prone.

Although lexical familiarity seems to be a crucial factor in L2 readability, only one LEXF feature appears in the top 30. All but two of the LEXF features that are based on the official Lexical Minimum or the Kelly Project appear near the bottom of the list, with less than 0.09 information gain. The purely frequency-based LEXF features, on the other hand, are more informative, with most showing at least 0.16 information gain. Of these, the standard deviation frequency features are the most informative.

The lexical complexity features have an average information gain of 0.166, with the highest values coming from various maximum word length features, for characters, syllables, and morphemes.

The morphology features have an average information gain of 0.157, with the most informative features including verbal morphology (participle, tense, person, etc.) and case distribution. The proportion of tokens that are parentheticals was also a very informative feature, which probably reflects the importance of parentheticals in higher level discourse.

The SENT features have an average information gain of 0.144, with subordinating conjunctions per sentence the highest.

The SYNT features have an average information gain of 0.279, with all but two features above 0.19.

The information gain metric does not account for informational overlap between features. In order to evaluate the usefulness of each feature as a member of a feature set, I used the correlation-based feature subset selection algorithm (CfsSubsetEval) (Hall, 1999), which selects the most predictive subset of features by minimizing redundant information, based on feature correlation. The results of CfsSubsetEval for the Combined corpus are given in Table 6.22, with features presented in alphabetical order.

Out of 179 features, the CfsSubsetEval algorithm selected 32 features. Many of the features selected for the optimal feature set are also among the top 30 most informative features in Table 6.21. However, the morphological features, which had only 7 features among the top 30, now include 14 features, which indicates that although these features are not as informative, the information that they contribute is unique.

The fact that the lexical familiarity features based on the lexical minimum vocabulary lists are most useful at the lower levels may indicate that determining which words a learner should know becomes increasingly difficult at higher levels.

A classifier trained on only these 32 features with the Combined corpus achieved

LEXC_CmaxWLchar
LEXC_maxWLmorph
LEXF_medContentFreq
LEXF_prctOverA1_lm
LEXF_prctOverA2_lm
LEXV_BTTR
LEXV_CTTR
LEXV_RTTR
LEXV_TlemTR
MORPH_ABBRperPOS
MORPH_CSPerWord
MORPH_PARTpresent
MORPH_PRNperPOS
MORPH_PastPerTense
MORPH_PresPerTense
MORPH_actvPerV
MORPH_indicPerV
MORPH_ppPerV
MORPH_pprsPerV
MORPH_prctIns
MORPH_prctNom
MORPH_prctPrt
MORPH_shrtPerA
SYNT_avgTreeDep
SYNT_avgmaxDepLen
SYNT_avgmaxTreeDep
SYNT_maxDepLen
SYNT_maxTreeDep
SYNT_maxVDeps
SYNT_maxavgDepLen
SYNT_maxavgTreeDep
TXT_dialogSymbCount

Table 6.22: 32 features selected by CfsSubsetEval, Combined corpus, all levels

precision 0.674 and recall 0.665 (F-score 0.659), which is only 0.01 worse than the model trained on all 180 features.

6.6.1 Feature evaluation with binary classifiers

In the learning curve experiment in Section 6.5.2 above, we saw that the two binary classifiers at the highest levels required the least training data. In order to investigate the cause of this outcome, I performed feature evaluation of binary datasets using the LQsupp subcorpus, the same corpus used for the learning curve experiment.

Using the correlation-based feature subset selection algorithm, I compared which features were selected for each binary dataset. The results of this analysis are given in Table 6.23.

A1-A2	A2-B1	B1-B2	B2-C1	C1-C2
DISC_Dial	LEXC_maxWLmorph	DISC_D/S	DISC_Dial	LEXC_prctContentLett12plus
DISC_PDtok	LEXF_prctOverA2_lm	LEXC_maxWLmorph	LEXC_prctContentLett4plus	LEXC_prctSyll4plus
LEXC_CmaxWLchar	MORPH_NPperPOS	LEXC_prctContentLett13plus	LEXF_avgFreq	LEXF_medContentFreq
LEXC_prctLett7plus	MORPH_actvPerV	LEXC_prctContentLett5plus	LEXF_avgTokFreqRank	LEXF_medTokFreq
LEXF_medContentFreq	MORPH_pprsPerV	LEXC_prctLett11plus	LEXF_medContentFreq	LEXF_prctOverA2_lm
LEXF_medContentTokFreq	MORPH_prctDat	LEXF_avgContentFreqRank	LEXF_medTokFreq	LEXF_stdevFreqRank
LEXF_prctOverA2_lm	SYNT_avgTreeDep	LEXF_prctOverA2_kp	LEXF_prctOverB1_kp	MORPH_ADVperPOS
LEXF_stdevFreq	SYNT_avgmaxDepLen	LEXF_prctOverB1_kp	LEXF_prctOverC1_kp	MORPH_FirstPerPers
MORPH_Abstr	SYNT_avgmaxNDeps	LEXF_stdevContentFreqRank	LEXV_CTTR	MORPH_PresPerTense
MORPH_DETperPOS	SYNT_maxDepLen	LEXF_stdevFreq	LEXV_RTTR	MORPH_ThirdPerPers
MORPH_NPperPOS	SYNT_maxavgTreeDep	LEXV_CTTR	MORPH_ABBRperPOS	MORPH_VBLEXperPOS
MORPH_SecondPerPers		LEXV_RTTR	MORPH_FirstPerPers	MORPH_infPerV
MORPH_VBLEXperPOS		MORPH_ADJperPOS	MORPH_NPperPOS	MORPH_prctAcc
MORPH_prctAcc		MORPH_Abstr	MORPH_NperPOS	MORPH_prctPrp
MORPH_prctDat		MORPH_DETperPOS	MORPH_ThirdPerPers	SENT_CS/S
SYNT_avgDepLen		MORPH_DETpresent	MORPH_prctIns	SENT_SLsyll
TXT_dialogSymbCount		MORPH_FirstPerPers	SENT_CLI	SYNT_maxavgNDeps
		MORPH_NUMperPOS	SYNT_avgmaxTreeDep	
		MORPH_PARENperPOS		
		MORPH_PRperPOS		
		MORPH_ThirdPerPers		
		MORPH_VBLEXperPOS		
		MORPH_actvPerV		
		MORPH_ppPerV		
		MORPH_prctDat		
		SENT_CS/S		
		SENT_SLchar		
		SYNT_avgDepLen		
		SYNT_maxTreeDep		
		SYNT_maxVDeps		
		SYNT_maxavgTreeDep		

Table 6.23: 32 features selected by CfsSubsetEval, Combined corpus, all levels

The most noticeable difference between the selected feature sets is that the B1-B2 feature set is much larger than the feature sets of other levels. This level also has the slowest learning curve above, and ultimately achieves lower accuracy. The two binary datasets at the higher complexity levels are dominated by lexical familiarity features (especially frequency-based features) and morphological features. Interestingly, the morphological features are primarily properties that reflect differences in register/genre, rather than morphological difficulty for language

learners. For example, whereas the morphological features of lower levels include prevalence of participles or passives, which are difficult for language learners, the morphology features of the higher levels reflect the prevalence of first-person or third-person verb forms, which do not themselves represent any kind of difficulty for language learners. The boosted importance of lexical familiarity features, and the diminished importance of grammatically "difficult" features in the higher binary datasets is supports the idea that language learning at level C is dominated by acquiring more specialized vocabulary, and finer points of usage, such as idiomatic collocations, phrases, and genre distinctions. This type of language development is more one-dimensional—from the perspective of feature extraction—and is therefore easier for our classifier to model with less data.

6.7 Conclusions and Outlook

This chapter has presented new research in automatic classification of Russian texts according to second language readability. This technology is intended to support learning activities that enhance student engagement through online authentic materials (Erbaggio et al., 2010). I collected a new corpus of Russian language-learning texts classified according to CEFR proficiency levels. The corpus comes from a broad spectrum of sources, which resulted in a richer and more robust dataset, while also complicating comparisons between subsets of the data. When training a model on one subcorpus, and testing on another, the results were consistently much lower than on cross-validation. This seems to indicate widespread disagreement over how to determine a text's readability. More consistent human ratings would result in better model portability between subcorpora. Given such low accuracy between subcorpora, it is not clear how well any of the models from this study will perform on new texts in real-world applications.

This chapter highlights the need to standardize how readability corpora are collected and validated. Despite the fact that readability assessment has been investigated in a large number of published studies, very little attention has been given to the prior, fundamental question of how well humans determine a given text's readability. The gold standard corpora used in most studies are texts that have been originally written for a target reading level, adapted for a target reading level, or categorized post hoc according to reading level. In each of these conditions, the readability of a text is being determined by a human—possibly with the assistance of readability formulas or models—and given the complexity of the task, there is no guarantee that a given human's ratings are reliable or valid. In fact, the results of this study suggest the opposite. As models of readability are becoming more sophisticated, it seems likely that computers can outperform humans at this task,

given a high-quality gold-standard corpus.

Future research is needed to identify methods to ensure higher quality ratings for gold standard readability corpora, possibly through post-hoc manual cleaning based on classifier "errors" or perhaps some other form of supervised bootstrapping. Psycholinguistic methods are well-suited to provide objective insights into text complexity. For example, two studies have investigated the effect of text complexity on eye tracking measures: (Rayner et al., 2006) and (Vajjala, 2015, ch. 4). Both of these studies identified eye-movement behaviors that correlated significantly with reported text difficulty, such as fixation count. Although the participants in these studies were native speakers, I expect that language learners would exhibit similar reactions to text complexity. This line of research shows promise for establishing objective grounds for building L2 readability corpora.

Classifier performance A six-level Random Forest classifier achieves an F-score of 0.671, with adjacent accuracy of 0.919. Binary classifiers with only two adjacent reading levels achieve F-scores between 0.806 and 0.892. This is the first large-scale study of this task with Russian data, and although these results are promising, there is still room for improvement, both in corpus quality and modeling features.

In Section 6.5.2, I found support for the hypothesis that simple texts are similar to one another and complex texts are complex in their own way by showing that binary classifiers at lower reading levels required less training data to approach their upper limit. With each successive binary classifier at progressively higher reading levels, the learning curve became slower, which I interpret as evidence that these levels can be difficult in many different ways. The two highest binary pairings are exceptional because the difference between adjacent levels is greater the higher you go.

Features Among the most informative individual features used in this study are type-token ratios (RTTR, CTTR, T_{lem}TR, TTR), as well as various measures of maximum syntactic dependency lengths and maximum tree depth. When features with overlapping information are removed, using correlation-based feature selection, we have 14 MORPH features, 8 SYNT features, 4 LEXV features, 3 LEXF features, and 2 LEXC features, and 1 DISC feature. Models trained on only one category of features also show the importance of morphology in this task, with the MORPH category achieving a higher F-score than other individual categories.

Although the feature set used in this study had fairly broad coverage, there are still a number of possible features that could likely improve classifier performance further. Other researchers have seen good results using features based on semantic

ambiguity, derived from word nets. Implementing such features would be possible with the new and growing resources from the Yet Another RussNet project.[20]

Another category of features that is absent in this study is language modeling, including the possibility of calculating information-theoretic metrics, such as surprisal, based on those models.

The syntactic features used in this study could be expanded to capture more nuanced features of the dependency structure. For instance, currently implemented syntactic features completely ignore the kinds of syntactic relations between words. In addition, some theoretical work in dependency syntax, such as catenae (Osborne et al., 2012) and dependency/locality (Gibson, 2000) may serve as the basis for other potential syntactic features.

Finally, it goes without saying that improving the performance of the morphological analyzer and constraint grammar on which the morphological features are based would lead to more precise morphological features that could, in turn, lead to more accurate classification.

Applications One of the most promising applications of the technology discussed in this chapter is a grammar-aware search engine or similar information retrieval framework that can assist both teachers and students to identify texts at the appropriate reading level. Such systems have been discussed in the literature (e.g. Ott, 2009), and similar tools can be created for Russian language learning.

[20] `http://russianword.net/en/`

Chapter 7

Conclusions and outlook

This dissertation is concerned with linguistic and computational analysis of Russian morphology in the context of language learning. In the preceding chapters, I have presented ground-breaking research with several interconnected technologies, all joined by the central task of supporting flexible, intelligent computer applications to help learners and teachers of Russian to find appropriate authentic texts, and automatically generate focus-on-form exercises from them. In broad terms, this dissertation has demonstrated the positive impact that automatic morphological analysis can contribute to Russian computer-assisted language learning.

7.1 Summary

In Chapter 2, I discussed both theoretical and practical issues surrounding the development of a two-level morphology of Russian. Many of the two-level rules were inspired by well-known research in the structure of Russian morphology, and other rules represent innovative approaches, unique to this grammar. The resulting finite-state morphological analyzer was shown to be competitive with state-of-the-art Russian morphological analyzers with respect to speed and coverage, while additionally meeting requirements specific to open language-learning applications; namely, it is free and open-source, it is designed to be wordstress-aware, and it serves as input to an efficient morphosyntactic disambiguation utility whose grammar can be designed to have high recall.

Chapter 3 presented a new constraint grammar for Russian. This grammar is designed to have high recall, which means that it only removes readings that can be ruled out with very high confidence (>99%). Despite the fact that Russian is a major world language, the major tagged corpora of Russian are not freely available. In order to work around this obstacle, we implemented an innovative development strategy for testing new rules using an untagged corpus, checking by

hand a sample of instances where the rule fires. This method is promising for languages with limited resources. We also demonstrated the utility of a high-recall grammar in a voting setup with a stochastic tagger—in this case a trigram tagger. This arrangement plays to the strengths of each model, achieving higher accuracy with less training data.

In Chapter 4, I presented ground-breaking research in automatic word stress placement. Placing stress in running text is crucial both for Russian language learners and for text-to-speech. applications. Although many research projects have reported on various approaches to resolving word stress ambiguity, or guessing the stress of unknown words, this is the first study to quantitatively evaluate the performance of the approach in running text. Our constraint grammar resolved 42% of the stress ambiguity in our corpus, and in combination with simple guessing strategies for unknown words, it achieved 96.15% accuracy, more than 6% improvement over the baseline dictionary lookup.

Chapter 5 demonstrated the use of the morphological analysis and disambiguation in an intelligent computer-assisted language learning application: Russian Visual Input Enhancement of the Web (RusVIEW). From an applied perspective, this application is perhaps the most exciting result of the dissertation, creating dynamic, interactive grammar excercises on important grammatical topics from virtually any online document. The morphological analyzer and constraint grammar facilitate the development of modules that can reliably target relevant wordforms, in part because of the high-recall nature of the grammar. The program "knows whether it knows", and can target forms that clear the confidence threshold for reliable grammar exercises. In addition, an important theoretical contribution of Chapter 5 is that it demonstrates the capability of a system based on native-language NLP to provide adaptive feedback, an ability that is frequently assumed to be possible only with specialized learner-language NLP.

The opportunities that are afforded by Russian VIEW are exciting, but they also make another difficulty that learners face more pronounced. How can a learner find texts that are at the appropriate reading level? The internet is an incomparable resource for authentic texts, but the majority of documents that a learner finds are too complex, which can lead to increased frustration and decreased confidence or motivation. Chapter 6 lays the foundation for overcoming this hurdle by developing classifier models to rate documents' readability according to the standard levels of the CEFR. I assembled a new Russian L2 readability corpus, and used a Random Forest classifier with 180 features describing each document's lexical variability, lexical familiarity, lexical complexity, morphological complexity/distribution, sentence/syntactic complexity, and discourse complexity. Using cross-validation, the Combined corpus achieved an F-score of 0.671 and adjacent accuracy of 0.919, whereas individual subcorpora achieved F-scores between 0.501 and 0.849 and ad-

jacent accuracy between 0.810 and 1.000.

With regard to which features are most important for readability classification, I showed that the morphology-based features provided by the technology developed in Chapters 2 and 3 are among the most informative (Karpov et al., 2014, cf.), and that among classifiers trained on only one category of features, the model trained on only morphological features outperformed the remaining categories. This supports the common-sense notion that morphology is one of the most difficult domains for Russian language learners, and it demonstrates the value of morphological analysis in Russian ATICALL applications.

7.2 Resources

Several resources were created in conjunction with this dissertation research, including NLP tools, corpora, and language-learning tools. The most notable resources from this dissertation are listed below. Unless otherwise noted, they are freely available and/or open source.

7.2.1 NLP tools

1. A Russian morphological analyzer/generator built as a two-level finite-state transducer. The transducer contains more than 100 000 lexemes with more than 2 000 000 surface-form/reading pairs. The wordforms in the transducer are marked with lexical stress, making it well-suited to language-learning applications.

2. A constraint grammar for Russian, designed to have high recall. The grammar contains 299 rules which remove 49% of ambiguity in running text.

7.2.2 Corpora

1. A small corpus of Russian with hand-disambiguated morphosyntactic annotation (about 10 000 tokens)

2. Corpus of Russian with marked lexical stress (7689 tokens). The corpus is representative of texts that learners of Russian are likely to encounter in their studies, including dialogs, prose, individual sentences that were written for learners, and excerpts from well-known literary works.

3. Second language readability corpus of Russian, with 4.3 million tokens and 4689 documents classified by the six CEFR aptitude levels: A1, A2, B1, B2, C1, C2. This corpus is by far the largest and most diverse corpus of its

kind for Russian. Many of the documents in the corpus are copyrighted, so it cannot be published openly. However, researchers who are interested in using the corpus for non-commercial purposes should contact me.

7.2.3 Language-learning tools

1. Russian activities on the VIEW framework, including the following grammar topics: noun inflection, adjective inflection, verb inflection, verbal aspect, participle formation, and word stress.

7.3 Outlook

Because this dissertation took a breadth-first approach to the issues surrounding Russian ICALL, the depth of each of these topics has not been explored completely, and much work remains to be done in each of these domains. In this section, I outline ways in which I anticipate building on the foundation that has been set by the research reported herein.

To begin with, the work of improving and maintaining the tools for morphological analysis and disambiguation will require continued research, including basic lexicography, developing new methods for guessing unknown wordforms, and improving the performance of our constraint grammar. Especially as the constraint grammar becomes more complete, these technologies can feed into higher-level analyses, such as syntactic parsing.

The word stress annotation project reported in Chapter 4 set a standard for empirical evaluation of the task, but was limited by many factors, including the size and quality of the gold corpus. Because this is a core function in Russian ATICALL, I plan to continue research on this task. In terms of experimental methodology, a much larger corpus has been collected and will soon be corrected by hand. Work is also continuing on improving the performance each component of the morphological analysis, morphosyntactic disambiguation, and unknown wordform guessing.

Many of the issues that arise during ICALL research are relevant to linguistic theory, and providing theoretically sound solutions to these questions can feed back into higher-quality ICALL products. For example, in developing a verbal aspect activity for Russian VIEW, the question arose how and whether particular tokens are constrained with respect to their aspect. In other words, how often is a verb's aspect determined by its syntactic context, and how often is it variable. The results of the corpus study reported in Chapter 5 seem to indicate that aspect is much more variable than I initially expected. Empirical linguistics research of this issue can therefore help determine which tokens in a text should be targeted for grammar

excercises, since verb tokens whose aspect is determined by context result in more reliable exercises that will not confuse a learner.

Finally, the opportunities made possible by Russian VIEW are inspired by theories in second language acquisition research which have yet to find sufficient support in empirical research, such as learning outcomes, increased motivation, etc. Because of the structured environment that VIEW creates, it is also an excellent testbed for these theories. In future research, I intend to implement these activities in structured experiments that may shed light on the efficacy of this approach.

7.4 Conclusion

This dissertation has resulted in the availability of free and open-source resources for Russian morphological analysis and disambiguation, with special emphasis on properties that are desirable for computer-assisted language-learning applications. Based on these technologies, I have established an empirical benchmark for automatic word stress annotation. I also developed intelligent computer-assisted language-learning activities on the VIEW platform, demonstrating their utility in this domain. Finally, I have laid the foundation for automatically classifying texts according to L2 reading level, which can save teachers time in identifying appropriate texts for their students, as well as increasing learner autonomy by allowing them to discover their texts for extracurricular reading and study.

The studies presented in this dissertation demonstrate the importance of Russian morphological analysis in computer-assisted language learning applications, and each study has also laid the foundation for continuing research in this sphere.

Bibliography

Aluisio, S., Specia, L., Gasperin, C., and Scarton, C. (2010). Readability assessment for text simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9.

Amaral, L. and Meurers, D. (2011). On Using Intelligent Computer-Assisted Language Learning in Real-Life Foreign Language Teaching and Learning. *ReCALL*, 23(1):4–24.

Antonsen, L., Wiechetek, L., and Trosterud, T. (2010). Reusing grammatical resources for new languages. In *Proceedings of the International conference on Language Resources and Evaluation LREC2010*, pages 2782–2789.

Beesley, K. R. and Karttunen, L. (2003). *Finite State Morphology*. CSLI Publications, Stanford.

Boguslavsky, I. and Iomdin, L. (2009). Semantics of attenuated comparatives in Russian. In *Proceedings [of the] Fourth International Conference on Meaning-Text Theory [Recurso electrónico]*, pages 65–76.

Boxarov, V., Alekseeva, S., Granovskij, D., Protopopova, E., Stepanova, M., and Surikov, A. (2013). Crowdsourcing morphological annotations. In Selegej, V., editor, *Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialog"*, volume 1.

Brants, T. (2000). TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 224–231.

Breidt, E. and Feldweg, H. (1997). Accessing foreign languages with COMPASS. *Machine Translation*, 12(1–2):153–174. Special Issue on New Tools for Human Translators.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brown, C., Snodgrass, T., Covington, M. A., Herman, R., and Kemper, S. J. (2007). Measuring propositional idea density through part-of-speech tagging. poster presented at Linguistic Society of America Annual Meeting, Anaheim, California.

Brown, C., Snodgrass, T., Kemper, S. J., Herman, R., and Covington, M. A. (2008). Automatic measurement of propositional idea density from part-of-speech tagging. *Behavior Research Methods*, 40(2):540–545.

Bybee, J. (2003). *Phonology and language use*, volume 94. Cambridge University Press.

Chall, J. S. and Dale, E. (1995). *Readability Revisted: The New Dale-Chall Readability Formula*. Brookline Books.

Church, K. (1985). Stress assignment in letter to sound rules for speech synthesis. *Association for Computational Linguistics*, pages 246–253.

Chvany, C. V. (1990). The two-stem nature of the one-stem verb system: Another look at classes and exceptions. *Slavic and East European Journal*, pages 421–438.

Cinman, L. and Sizov, V. (2000). Lingvističeskij processor etap: deskriptornoe sootvetstvie i obrabotka metafor [linguistic processor etap: descriptor correspondence and processing metaphors]. In *Trudy mežđunarodnogo seminara Dialog 2000 [Proceedings of the international seminar Dialog 2000]*, pages 366–369, Moscow. RGGU [Russian State University for the Humanities].

Coleman, M. and Liau, T. L. (1975). A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60:283–284.

Collins-Thompson, K. and Callan, J. (2004). A language modeling approach to predicting reading difficulty. In *Proceedings of HLT/NAACL 2004*, Boston, USA.

Comer, W. J. and deBenedette, L. (2011). Processing Instruction and Russian: Further evidence is IN. *Foreign Language Annals*, 44(4):646–673.

Crossley, S. A., Dufty, D. F., McCarthy, P. M., and McNamara, D. S. (2007a). Toward a new readability: A mixed model approach. In McNamara, D. S. and Trafton, G., editors, *Proceedings of the 29th annual conference of the Cognitive Science Society*. Cognitive Science Society.

Crossley, S. A., Greenfield, J., and McNamara, D. S. (2008). *Assessing Text Readability Using Cognitively Based Indices*, pages 475–493. Teachers of English

to Speakers of Other Languages, Inc. 700 South Washington Street Suite 200, Alexandria, VA 22314.

Crossley, S. A., Louwerse, M. M., McCarthy, P. M., and McNamara, D. S. (2007b). A linguistic analysis of simplified and authentic texts. *The Modern Language Journal*, 91(1):15–30.

Crosswhite, K., Alderete, J., Beasley, T., and Markman, V. (2003). Morphological effects on default stress in novel Russian words. In *WCCFL 22 Proceedings*, pages 151–164.

Čukovskij, K. I. (2009). *Živoj kak žizn' [Alive as life]*. Zebra-E.

Dale, E. and Chall, J. S. (1948). A formula for predicting readability. *Educational research bulletin; organ of the College of Education*, 27(1):11–28.

Dell'Orletta, F., Montemagni, S., and Venturi, G. (2011). Read-it: Assessing readability of italian texts with a view to text simplification. In *Proceedings of the 2nd Workshop on Speech and Language Processing for Assistive Technologies*, pages 73–83.

Dickinson, M. (2010). Generating learner-like morphological errors in Russian. In *Proceedings of the 23nd International Conference on Computational Linguistics (COLING-10)*, Beijing.

Dickinson, M. and Herring, J. (2008a). Developing online ICALL exercises for Russian. In *The 3rd Workshop on Innovative Use of NLP for Building Educational Applications (BEA-3) at ACL'08*, pages 1–9, Columbus, OH.

Dickinson, M. and Herring, J. (2008b). Russian morphological processing for ICALL. In *The Fifth Midwest Computational Linguistics Colloquium (MCLC-5)*, East Lansing, MI.

Dou, Q., Bergsma, S., Jiampojamarn, S., and Kondrak, G. (2009). A ranking approach to stress prediction for letter-to-phoneme conversion. In *Proceedings of the Joint Conference of the 47th annual meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 118–126, Suntec, Singapore. Association for Computational Linguistics.

DuBay, W. H. (2006). *The Classic Readability Studies*. Impact Information, Costa Mesa, California.

Ellis, R. (2009). A typology of written corrective feedback types. *ELT journal*, 63(2):97–107.

Endresen, A., Janda, L., Reynolds, R., and Tyers, F. (2016). Who needs particles? a challenge to the classification of particles as a part of speech in russian. *Russian Linguistics*, 40(2).

Erbaggio, P., Gopalakrishnan, S., Hobbs, S., and Liu, H. (2010). Enhancing student engagement through online authentic materials. *International Association for Language Learning Technology*, 42(2).

Erjavec, T. (2004). Multext-east version 3: Multilingual morphosyntactic specifications, lexicons and corpora. In *LREC*.

Ezeiza, N., Alegria, I., Arriola, J. M., Urizar, R., and Aduriz, I. (1998). Combining stochastic and rule-based methods for disambiguation in agglutinative languages. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 380–384. Association for Computational Linguistics.

Feng, L. (2010). *Automatic Readability Assessment*. PhD thesis, City University of New York (CUNY).

Feng, L., Jansche, M., Huenerfauth, M., and Elhadad, N. (2010). A comparison of features for automatic readability assessment. In *In Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Beijing, China.*

Filippova, A. V. (2010). *Upravlenie kačestvom učebnyx materialov na osnove analize trudnosti ponimanija učebnyx tekstov [Managing the quality of educational materials on the basis of analyzing the difficulty of understanding educational texts]*. PhD thesis, Ufa State Aviation Technology University.

Francois, T. and Watrin, P. (2011). On the contribution of mwe-based features to a readability formula for french as a foreign language. In *Proceedings of Recent Advances in Natural Language Processing*, pages 441–447.

Gal', N. (2014). *Slovo živoe i mërtvoe [Word alive and dead]*. Vremja.

Gibson, E. (2000). The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, pages 95–126.

Gilmore, A. (2007). Authentic materials and authenticity in foreign language learning. *Language teaching*, 40(02):97–118.

Giménez, J. and Màrquez, L. (2004). Svmtool: A general pos tagger generator based on support vector machines. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation*.

Gorisev, S., Koynov, A., Kuzemchik, V., Lisinin, S., Mikhaleva, E., Mishunin, O., Savinov, A., Terekhin, D., Firstov, D., and Cherkashin, A. (2013). Intellektual'nyj lingvoprocessornyj kompleks "klios" dl'a obučenija rki [Intelligent language-aware tutoring system "KLIOS" for studying Russian as a foreign language]. *Sovremennye problemy nauki i obrazovanija [Modern problems of science and education]*, 6.

Graesser, A. C., McNamara, D. S., Louweerse, M. M., and Cai, Z. (2004). Coh-metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments and Computers*, 36:193–202.

Grishina, E. and Lyashevskaya, O. (2008). *Grammatičeskij slovar' novyx slov russkogo jazyka [Grammatical dictionary of new Russian words]*. `http://dict.ruslang.ru/gram.php` [Accessed: Jan 2015].

Hajič, J., Krbec, P., Květoň, P., Oliva, K., and Petkevič, V. (2001). Serial combination of rules and statistics: A case study in czech tagging. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 268–275. Association for Computational Linguistics.

Hajič, J., Votrubec, J., Krbec, P., Květoň, P., et al. (2007). The best of two worlds: Cooperation of statistical and rule-based taggers for czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*, pages 67–74. Association for Computational Linguistics.

Hall, J., Nilsson, J., and Nivre, J. (2009). *Homepage of MaltParser*.

Hall, K. and Sproat, R. (2013). Russian stress prediction using maximum entropy ranking. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 879–883, Seattle, Washington, USA. Association for Computational Linguistics.

Hall, M. A. (1999). *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato.

Halle, M. (1994). The Russian declension: An illustration of the theory of Distributed Morphology. *Perspectives in Phonology. CSLI Publications, Stanford*, pages 29–60.

Halácsy, P., Kornai, A., and Oravecz, C. (2007). Hunpos: An open-source trigram tagger. In *Proceedings of the 45th annual meeting of the ACL*, pages 209–212.

Hancke, J., Meurers, D., and Vajjala, S. (2012). Readability classification for german using lexical, syntactic, and morphological features. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 1063–1080, Mumbay, India.

Heilman, M., Collins-Thompson, K., Callan, J., and Eskenazi, M. (2007). Combining lexical and grammatical features to improve readability measures for first and second language texts. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL-07)*, pages 460–467, Rochester, New York.

Heilman, M., Collins-Thompson, K., and Eskenazi, M. (2008a). An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the 3rd Workshop on Innovative Use of NLP for Building Educational Applications at ACL-08*, Columbus, Ohio.

Heilman, M., Zhao, L., Pino, J., and Eskenazi, M. (2008b). Retrieval of reading materials for vocabulary and reading practice. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications (BEA-3) at ACL'08*, pages 80–88, Columbus, Ohio.

Hulden, M. and Francom, J. (2012). Boosting statistical tagger accuracy with simple rule-based grammars. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*.

Ivanov, V. V. (2013). K voprocu o vozmožnosti ispol'zovanija lingvističeskix xarakteristik složnosti teksta pri issledovanii okulomotornoj aktivnosti pri čtenii u podrostkov [toward using linguistic profiles of text complexity for research of oculomotor activity during reading by teenagers]. *Novye issledovanija [New studies]*, 34(1):42–50.

Janda, L. (2007). Aspectual clusters of Russian verbs. *Studies in language*, 31(3):607–648.

Janda, L. A. and Lyashevskaya, O. (2011). Aspectual pairs in the russian national corpus. *Scando-Slavica*, 57(2):201–215.

Johannessen, J. B., Hagen, K., Lynum, A., and Nøklestad, A. (2011). OBT+Stat: Evaluation of a combined CG and statistical tagger. In Bick, E., Hagen, K., Müürisep, K., and Trosterud, T., editors, *Proceedings of the NODALIDA 2011*

Workshop Constraint Grammar Applications, volume 14, pages 26–34, Riga, Latvia. NEALT.

Johannessen, J. B., Hagen, K., Lynum, A., and Nøklestad, A. (2012). Obt+stat: A combined rule-based and statistical tagger. In Andersen, G., editor, *Exploring Newspaper Language: Using the Web to Create and Investigate a Large Corpus of Modern Norwegian*, pages 51–66. John Benjamins Publishing.

Karlsson, F. (1990). Constraint grammar as a framework for parsing running text. In *Proceedings of the 13th Conference on Computational Linguistics (COLING), Volume 3*, pages 168–173, Helsinki, Finland. Association for Computational Linguistics.

Karlsson, F., Voutilainen, A., Heikkilä, J., and Anttila, A., editors (1995). *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Number 4 in Natural Language Processing. Mouton de Gruyter, Berlin and New York.

Karpov, N., Baranova, J., and Vitugin, F. (2014). Single-sentence readability prediction in Russian. In *Proceedings of Analysis of Images, Social Networks, and Texts conference (AIST)*.

Kilgarriff, A., Charalabopoulou, F., Gavrilidou, M., Johannessen, J. B., Khalil, S., Kokkinakis, S. J., Lew, R., Sharoff, S., Vadlapudi, R., and Volodina, E. (2014). Corpus-based vocabulary lists for language learners for nine languages. *Language resources and evaluation*, 48(1):121–163.

Kincaid, J. P., Fishburne, R. P. J., Rogers, R. L., and Chissom, B. S. (1975). Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease formula) for Navy enlisted personnel. Research Branch Report 8-75, Naval Technical Training Command, Millington, TN.

Kopotev, M. and Mustajoki, A. (2003). Principy sozdanija Xel'sinkskogo annotirovannogo korpusa russkix tekstov (HANCO) v seti Internet [principles of the creation of the Helsinki annotated corpus of Russian texts]. *Naučno-texničeskaja informacija*, 2:33–36.

Korobov, M. (2015). Morphological analyzer and generator for Russian and Ukrainian languages. In *Proceedings of AIST'2015*. Springer.

Koskenniemi, K. (1983). Two-level morphology: A general computational model for word-form recognition and production. Technical report, University of Helsinki, Department of General Linguistics.

Koskenniemi, K. (1984). A general computational model for word-form recognition and production. In *Proceedings of the 10th International Conference on Computational Linguistics*, COLING '84, pages 178–181, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kotlyarov, A. (2015). Measuring and analyzing comprehension difficulty of texts in contemporary Russian. In *Materials of the annual scientific and practical conference of students and young scientists (with international participation)*, pages 63–65, Kostanay, Kazakhstan.

Krashen, S. (1977). Some issues relating to the monitor model. *On Tesol*, 77(144-158).

Krashen, S. D. (1985). *The input hypothesis: Issues and implications*. Addison-Wesley Longman Ltd.

Krashen, S. D. and Terrell, T. D. (1983). *The natural approach: Language acquisition in the classroom.* Pergamon.

Krioni, N. K., Nikin, A. D., and Filippova, A. V. (2008). Avtomatizirovannaja sistema analiza složnosti učebnyx tekstov [automated system for analyzing the complexity of educational texts]. *Vestnik Ufimskogo gosudarstvennogo aviacionnogo texničeskogo universiteta [Bulletin of the Ufa State Aviation Technical University]*, 11(1):101–107.

Krivnova, O. (1998). Avtomatičeskij sintez russkoj reči po proizvol'nomu tekstu (vtoraja versija s ženskim golosom) [Automatic Russian speech synthesis with unrestricted text (version 2 with female voice)]. In *Trudy meždunarodnogo seminara Dialog [Proceedings of the international seminar Dialog]*, pages 498–511.

Krylov, S. and Starostin, S. (2003). Aktual'nye zadači morfologičeskogo analiza i sinteza v integrirovannoj informacionnoj srede STARLING [upcoming tasks for morphological analysis and generation in the integrated information environment STARLING]. In *Proceedings of the International Conference "Dialog 2003"*.

Kuznetsova, J. (2013). *Linguistic Profiles: Correlations between Form and Meaning*. PhD thesis, University of Tromsø. Doctoral Dissertation.

Lavitskaya, Y. and Kabak, B. (2014). Phonological default in the lexical stress system of Russian: Evidence from noun declension. *Lingua*, 150:363–385.

Leaver, B. L., Rifkin, B., and Shekhtman, B. (2004). Apples and oranges are both fruit, but they don't taste the same: A response to Wynne Wong and Bill Van Patten. *Foreign Language Annals*, 37(1):125–132.

Lee, S.-K. and Huang, H.-T. (2008). Visual Input Enhancement and Grammar Learning: A meta-analytic review. *Studies in Second Language Acquisition*, 30:307–331.

Leow, R. P. (2009). Input enhancement and l2 grammatical development: What the research reveals. In Katz, S. L. and Watzinger-Tharp, J., editors, *Conceptions of L2 Grammar: Theoretical Approaches and their Applications in the L2 Classroom*, AAUSC 2008 Volume, pages 16–34. Heinle Cengage Learning.

Linden, K., Silfverberg, M., Axelson, E., Hardwick, S., and Pirinen, T. (2011). Hfst—framework for compiling and applying morphologies. In Mahlow, C. and Pietrowski, M., editors, *Systems and Frameworks for Computational Morphology*, volume Vol. 100 of *Communications in Computer and Information Science*, pages 67–85. Springer.

Ljaševskaja, O. N. and Šarov, S. (2009). *Častotnyj slovar' sovremennogo russkogo jazyka (na materialax Nacional'nogo korpusa russkogo jazyka) [Frequency dictionary of Modern Russian (based on the Russian National Corpus)]*. Azbukovnik, Moscow.

Long, M. H. (1981). Input, interaction, and second-language acquisition. *Annals of the New York Academy of Sciences*, 379(1):259–278.

Long, M. H. (1983). Does second language instruction make a difference? a review of research. *Tesol Quarterly*, pages 359–382.

Lu, X. (2012). The relationship of lexical richness to the quality of ESL learners' oral narratives. *The Modern Languages Journal*, pages 190–208.

Lyster, R. and Ranta, L. (1997). Corrective feedback and learner uptake: Negotiation of form in communicative classrooms. *Studies in Second Language Acquisition*, 19 n1:37–66.

Meurers, D. (2012). Natural language processing and language learning. In Chapelle, C. A., editor, *Encyclopedia of Applied Linguistics*, pages 4193–4205. Wiley, Oxford.

Meurers, D., Ziai, R., Amaral, L., Boyd, A., Dimitrov, A., Metcalf, V., and Ott, N. (2010). Enhancing authentic web pages for language learners. In *Proceedings of the 5th Workshop on Innovative Use of NLP for Building Educational Applications (BEA-5) at NAACL-HLT 2010*, pages 10–18, Los Angeles.

Mikk, J. A. (1974). Metodika razrabotki formul čitabel'nosti [methods for developing readability formulas]. *Sovetskaja pedagogika i škola IX*, page 273.

Mizernov, I. J. and Graščenko, L. A. (2015). Analiz metodov ocenki složnosti teksta [analysis of methods for evaluating text complexity]. *Novye informacionnye texnologii v avtomatizirovannyx sistemax [New information technologies in automated systems]*, 18:572–581.

Morrow, K. (1977). Authentic texts in esp. *English for specific purposes*, pages 13–16.

Nerbonne, J. (2003). Natural language processing in computer-assisted language learning. In Mitkov, R., editor, *The Oxford Handbook of Computational Linguistics*. Oxford University Press.

Nerbonne, J., Dokter, D., and Smit, P. (1998). Morphological processing and computer-assisted language learning. *Computer Assisted Language Learning*, 11(5):543–559.

Nesset, T. (2008). *Abstract phonology in a concrete model: Cognitive linguistics and the morphology-phonology interface*. Walter de Gruyter.

Nikin, A. D., Krioni, N. K., and Filippova, A. V. (2007). Informacionnaja sistema analiza učebnogo teksta [information system for analyzing educational texts]. In *Trudy XIV Vserossijskoj naučno-metodičskoj konferencii Telematika [Proceedings of the XIV pan-Russian scientific-methodological conference Telematika]*, pages 463–465.

Nozhov, I. (2003). *Morfologičeskaja i sintaksičeskaja obrabotka teksta (modeli i programmy) [Morphological and Syntactic Text Processing (models and programs)] also published as Realizacija avtomatičeskoj sintaksičeskoj segmentacii russkogo predloženija [Realization of automatic syntactic segmentation of the Russian sentence]*. PhD thesis, Russian State University for the Humanities, Moscow.

Oborneva, I. V. (2005). Matematičeskaja model' ocenki učebnyx tekstov [mathematical model of evaluation of scholastic texts]. In *Informacionnye texnologii v obrazovanii: XV Meždunarodaja konferencija-vystavka [Information technology in education: XV international conference-exhibit*.

Oborneva, I. V. (2006a). Avtomatizacija ocenki kačestva vosprijatija teksta [automation of evaluating the quality of text comprehension]. No longer available on internet.

Oborneva, I. V. (2006b). *Avtomatizirovannaja ocenka složnosti učebnyx tekstov na osnove statističeskix parametrov [Automatic evaluation of the complexity of educational texts on the basis of statistical parameters]*. PhD thesis.

Oflazer, K. and Tür, G. (1996). Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation. In *Proceedings of the ACLSIGDAT Conference on Empirical Methods in Natural Language Processing*, pages 69–81, Philadelphia, PA, USA.

Okladnikova, S. V. (2010). Model' kompleksnoj ocenki čitabel'nosti testovyx materialov na etape razrabotki [a model of multidimensional evaluation of the readability of test materials at the development stage]. *Prikaspijskij žurnal: upravlenie i vysokie texnologii*, 3:63–71.

Osborne, T., Putnam, M., and Groß, T. (2012). Catenae: Introducing a novel unit of syntactic analysis. *Syntax*, 15(4):354–396.

Ott, N. (2009). Information retrieval for language learning: An exploration of text difficulty measures. ISCL master's thesis, Universität Tübingen, Seminar für Sprachwissenschaft, Tübingen, Germany.

Ott, N. and Meurers, D. (2010). Information retrieval for education: Making search engines language aware. *Themes in Science and Technology Education. Special issue on computer-aided language analysis, teaching and learning: Approaches, perspectives and applications*, 3(1–2):9–30.

Pearson, S., Kuhn, R., Fincke, S., and Kibre, N. (2000). Automatic methods for lexical stress assignment and syllabification. In *International Conference on Spoken Language Processing*, pages 423–426.

Peradin, H. and Šnajder, J. (2012). Towards a constraint grammar based morphological tagger for croatian. In *Proceedings of the 15th International Conference, TSD 2012*, pages 174–182, Brno. Springer.

Petersen, S. E. and Ostendorf, M. (2009). A machine learning approach to reading level assessment. *Computer Speech and Language*, 23:86–106.

Petrova, I. J. and Okladnikova, S. V. (2009). Metodika rasčeta bazovyx pokazatelej čitabel'nosti testovyx materialov na osnove ekspertnyx ocenok [method of calculating basic indicators of readability of test materials on the basis of expert evaluations]. *Prekaspijskij žurnal: upravlenie i vysokie texnologii*, page 85.

Pilán, I., Vajjala, S., and Volodina, E. (2015). A readable read: Automatic assessment of language learning materials based on linguistic complexity. In *Proceedings of CICLING 2015- Research in Computing Science Journal Issue (to appear)*.

Polio, C. (2007). A history of input enhancement: Defining an evolving concept. *Assessing the impact of input enhancement in second language acquisition: Evolution in theory, research and practice*, pages 1–18.

Presson, N., Davy, C., and MacWhinney, B. (2013). Experimentalized call for adult second language learners. In Schwieter, J. W., editor, *Innovative Research and Practices in Second Language Acquisition and Bilingualism*, pages 139–164. John Benjamins.

Rayner, K., Chace, K. H., Slattery, T. J., and Ashby, J. (2006). Eye movements as reflections of comprehension processes in reading. *Scientific Studies of Reading*, 10(3):241–255.

Reynolds, R., Schaf, E., and Meurers, D. (2014). A view of Russian: Visual input enhancement and adaptive feedback. In *NEALT Proceedings Series*, volume 22, pages 98–112, Uppsala.

Reynolds, R. and Tyers, F. (2015). Automatic word stress annotation of Russian unrestricted text. In *Main conference proceedings from NODALIDA 2015*, Vilnius, Lithuania. NEALT.

Robinson, P., Mackey, A., Gass, S., and Schmidt, R. (2012). Attention and awareness in second language acquisition. *The Routledge handbook of second language acquisition*, pages 247–267.

Roll, M., Frid, J., and Horne, M. (2007). Measuring syntactic complexity in spontaneous spoken Swedish. *Language and Speech*, 50(2):227–245.

Rutherford, W. E. and Sharwood Smith, M. (1985). Consciousness-raising and universal grammar. *Applied Linguistics*, 6(2):274–282.

Schmid, H. (2004). Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland.

Schmidt, R. (1990). The role of consciousness in second language learning. *Applied Linguistics*, 11:206–226.

Schmidt, R. (2010). Attention, awareness, and individual differences in language learning. In Chan, W. M., Chi, S., Cin, K. N., Istanto, J., Nagami, M., Sew, J. W., Suthiwan, T., and Walker, I., editors, *Proceedings of CLaSIC 2010, Singapore, December 2-4*, pages 721–737, Singapore. National University of Singapore, Centre for Language Studies.

Schwarm, S. and Ostendorf, M. (2005). Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 523–530, Ann Arbor, Michigan.

Segalovich, I. (2003). A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In *International Conference on Machine Learning; Models, Technologies and Applications*, pages 273–280.

Sharoff, S., Kopotev, M., Erjavec, T., Feldman, A., and Divjak, D. (2008a). Designing and evaluating a Russian tagset. In *Proceedings of LREC*, Marrakech.

Sharoff, S., Kurella, S., and Hartley, A. (2008b). Seeking needles in the web's haystack: Finding texts suitable for language learners. In *Proceedings of the 8th Teaching and Language Corpora Conference (TaLC-8)*, Lisbon, Portugal.

Sharoff, S. and Nivre, J. (2011). The proper place of men and machines in language technology: Processing Russian without any linguistic knowledge. In *Komp'uternaja lingvistika i intellektual'nye texnologii: Po materialam Mezhdunarodnoj konferencii "Dialog"*, pages 591–604, Bekasovo, Russia.

Sharwood Smith, M. (1981). Consciousness-raising and the second language learner. *Applied Linguistics*, 11(2):159–179.

Sharwood Smith, M. (1991). Speaking to many minds: On the relevance of different types of language information for the L2 learner. *Second Language Research*, 7(2):118–132.

Sharwood Smith, M. (1993). Input enhancement in instructed SLA: Theoretical bases. *Studies in Second Language Acquisition*, 15:165–179.

Sharwood Smith, M. (2014). Possibilities and limitations of enhancing language input: a mogul perspective. In Benati, A., Laval, C., and Arche, M. J., editors, *The grammar dimension in instructed second language learning*, pages 37–57. Bloomsbury Academic, London.

Si, L. and Callan, J. (2001). A statistical model for scientific readability. In *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM)*, pages 574–576. ACM.

Sokirko, A. and Toldova, S. (2004). Sravnenie effektivnosti dvux metodik snjatija leksičeskoj i morfologičeskoj neodnoznačnosti dlja russkogo jazyka (skrytaja model' markova i sintaksičeskij analizator imennyx grupp) [comparing the effectivness of two methods of removing lexical and morphological ambiguity for Russian (Hidden Markov Model and syntactic analysis of nominal groups)]. In *Meždunarodnaja Konferencija "Korpusnaja lingvistika 2004" [International conference "Corpus linguistics 2004"]*, St. Petersburg.

Špakovskij, J. F. (2003). *Formuly čitabel'nosti kak metod ocenki kačestva knigi [Formulae of readability as a method of evaluating the quality of a book]*, pages 39–48. Ukrainska akademija drukarstva, Lviv'.

Špakovskij, J. F. (2008). Razrabotka količestvennoj metodiki ocenki trudnosti vosprijatija učebnyx tekstov dl'a vyssšej školy [development of quantitative methods of evaluating the difficulty of comprehension of educational texts for high school]. *Naučno-texničeskij vestnik [Instructional-technology bulletin]*, pages 110–117.

Stenner, A. J. (1996). Measuring reading comprehension with the lexile framework. In *Fourth North American Conference on Adolescent/Adult Literacy*.

Swain, M. (1985). Communicative competence: Some roles of comprehensible input and comprehensible output in its development. *Input in second language acquisition*, 15:165–179.

Swain, M. (2005). The output hypothesis: Theory and research. In Hinkel, E., editor, *Handbook on research in second language teaching and learning*, pages 471–484. Lawrence Erlbaum Associates, Mahwah, NJ.

Tixonov, A. N. (2002). *Morfemno-orfografičeskij slovar': okolo 100 000 slov [Morpho-orthographic dictionary: apprx 100 000 words]*. AST/Astrel', Moskva.

Truscott, J. (1996). The case against grammar correction in l2 writing classes. *Language learning*, 46(2):327–369.

Tyers, F. and Reynolds, R. (2015). A preliminary constraint grammar of Russian. In *Proceedings of the workshop on "Constraint Grammar — methods, tools and applications"*, pages 39–46, Vilnius. Linköping University Electronic Press.

Vajjala, S. (2015). *Analyzing Text Complexity and Text Simplification: Connecting Linguistics, Processing and Educational Applications*. PhD thesis, University of Tübingen.

Vajjala, S. and Meurers, D. (2012). On improving the accuracy of readability classification using insights from second language acquisition. In Tetreault, J., Burstein, J., and Leacock, C., editors, *In Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–173, Montréal, Canada. Association for Computational Linguistics.

Vilkki, L. (1997). RUSTWOL: A system for automatic recognition of Russian words. Technical report, Lingsoft, Inc.

Vilkki, L. (2005). RUSTWOL: A tool for automatic Russian word form recognition. In Arppe, A., Carlson, L., Lindén, K., Piitulainen, J., Suominen, M., Vainio, M., Westerlund, H., and Yli-Jyrä, A., editors, *Inquiries into Words, Constraints and Contexts: Festschrift for Kimmo Koskenniemi on his 60th Birthday*, pages 151–162. CSLI Publications.

Vor der Brück, T. and Hartrumpf, S. (2007). A semantically oriented readability checker for german. In Vetulani, Z., editor, *Proceedings of the 3rd Language & Technology Conference*, pages 270–274, Poznań, Poland. Wydawnictwo Poznańskie.

Vor der Brück, T., Hartrumpf, S., and Helbig, H. (2008). A readability checker with supervised learning using deep syntactic and semantic indicators. *Informatica*, 32(4):429–435.

Voutilainen, A. (2004). Hand crafted rules. In van Halteren, H., editor, *Syntactic Wordclass Tagging*, pages 217–246. Kluwer Academic.

Vygotsky, L. S. (1986). *Thought and Language*. MIT Press, Cambridge, MA.

Webster, G. (2004). Improving letter-to-pronunciation accuracy with automatic morphologically-based stress prediction. In *Eighth International Conference on Spoken Language Processing*, pages 2573–2576.

Williams, B. (1987). Word stress assignment in a text-to-speech synthesis system for british english. *Computer Speech and Language*, 2:235–272.

Wong, W. and Van Patten, B. (2003). The evidence is IN: Drills are OUT. *Foreign Language Annals*, 36(3):403–423.

Wong, W. and Van Patten, B. (2004). Beyond experience and belief (or, Waiting for the evidence): A reply to Leaver et al.'s "Apples and oranges". *Foreign Language Annals*, 37(1):133–142.

Wądołowska-Lesner, K. (2011). Stepen' jazykovoj trudnosti russkix didaktičeskix tekstov [degree of linguistic difficulty of Russian didactic texts]. In *Sborník příspěvků z mezinárodní konference Rossica Olomucensia L*, pages 107–109, Olomouc.

Xomicevič, O., Rybin, S., Talanov, A., and Oparin, I. (2008). Avtomatičeskoe opredelenie mesta udarenie v neznakomyx slovax v sisteme sinteza reči [Automatic determination of the place of stress in unknown words in a speech synthesis system]. In *Materialy XXXVI meždunarodnoj filologičeskoj konferencii [Proceedings of the XXXVI International Philological Conference]*, Saint Petersburg.

Yannakoudakis, H., Briscoe, T., and Medlock, B. (2011). A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 180–189, Stroudsburg, PA, USA. Association for Computational Linguistics. Corpus available: `http://ilexir.co.uk/applications/clc-fce-dataset`.

Zaliznjak, A. A. (1977). *Grammatičeskij slovar' russkogo jazyka: slovoizmenenie: okolo 100 000 slov [Grammatical dictionary of the Russian language: Inflection: approx 100 000 words]*. Russkij jazyk.

Zelenkov, J., Segalovič, I., and Titov, V. (2005). Verojatnostnaja model' snjatija morfologičeskoj omonimii na osnove normalizujuščix podstanovok i pozicij sosednix slov [probabilistic model for removing morphological homonymy based on normalizing substitutions and positions of neighboring words]. In *Komp'juternaja lingvistika i intellektual'nye texnologii. Trudy meždunarodnogo seminara Dialog [Computational linguistics and intellectual technologies. Proceedings of the international seminar Dialog]*, pages 188–197.