Building a Tagged Corpus of Russian: A Bazaar Approach

Christopher Tessone
ctessone@knox.edu

11th June 2004

A THESIS SUBMITTED TO THE DEPARTMENT OF MODERN LANGUAGES OF KNOX COLLEGE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ARTS

Committee: Charles Mills, Visiting Instructor of Russian, Knox College
Don Blaheta, Assistant Professor of Computer Science, Knox College
Gerald Krumbholz, Visiting Assistant Professor of Music, Knox College
Steven Clancy, Senior Lecturer in Slavic, University of Chicago

Building a Tagged Corpus of Russian: A Bazaar Approach

Christopher Tessone
ctessone@knox.edu

11th June 2004

Abstract

This paper describes the development of a tagged corpus of Russian text and a suite of corpus development tools for use by researchers and programmers in the field of natural language processing (NLP). Few tools and corpora are freely available for commercial use, thus removing the possibility of spreading the annotation burden across many potential users of the corpus as open-source project managers have done with programming projects for years. We propose a new model for corpus development that involves potential users of corpus data outside the academic sphere and relieves the financial and labor burdens placed on institutions that develop corpora. We also present several strategies for speeding up the annotation process.

1 Introduction

Recent years have seen a tremendous growth in the field of corpus linguistics, largely due to the advent of increasingly fast computers with greater storage capacity. With the appearance of large, computer-searchable corpora—databases of natural-language text that include information about the text, like part-of-speech and morphological markers—the natural language processing (NLP) community has turned to statistical methods for exploiting annotated corpora to improve machine translation systems, question answerers, speech generation and recognition systems, and so on. Linguists have used them for everything from demonstrating the relative rarity of the subject–intransitive verb sentence type in English (Sampson 2003) to teaching modal particles in German (Möllering 2001).

However, very little work has been done in applying statistical methods to Russian or improving Russian language processing systems using results

from NLP. This is largely due to a shortage of annotated corpora of Russian. TeCoRus, a corpus of recorded speech from telephone conversations, has limited use for general NLP research because speech is so different from written texts, and the remaining corpora (including the Uppsala Corpus of Russian Texts from Uppsala University, Sweden, and the Corpus of Interviews from Tübingen University, Germany, both of which use mostly newspaper texts and novels) contain almost no annotation and are little better than the mass of text already available on the websites of dozens of Russian newspapers.[1] Annotated corpora are still in early stages of development and do not address some very large spheres of Russian usage.[2] In particular, they do not address a growing variety of written Russian: that of the Russian-language Internet. The register of Russian used there is somewhere between speech and standard written Russian. Because the text has not been edited by several people as most newspaper texts are, it is very useful for software which accepts user-written input, which is likely to be closer to this new register of the language.

In this paper, we describe the construction of a new corpus of Russian and propose several strategies for reducing the considerable labor required to develop a corpus from unannotated text files. In addition to the corpus and methodological suggestions, the project has also produced a suite of open-source software for developing corpora. The results suggest that, with the right software, it is possible for one person to develop a small corpus and in twenty hours train a part-of-speech tagger that will perform with an accuracy of 75–78%. Potential applications of this project include developing an open-source morphological analyzer for Russian and providing data for use by linguists working with Russian.

2 Project Overview

In October of 2003 and April of 2004, we asked members of the Russian-language community of the LiveJournal website (`http://www.livejournal.`

[1] Information about TeCoRus is available at `http://www.ccas.ru/depart/chuchu/doc_en/projects/Tecorus_info_en.pdf`. The Uppsala and Tübingen corpora are available at `http://www.sfb441.uni-tuebingen.de/b1/en/korpora.html`.

[2] At the time this thesis was defended, a group of scholars working in Russian, computer science, and linguistics released the National Corpus of Russian, found at `http://www.ruscorpora.ru/`. Although the corpus contains only 20,000,000 tokens at the time of writing, it is a significant event in Russian corpus linguistics. However, it still does not address Internet usage, and the licensing is very restrictive, permitting only non-commercial academic usage.

com/) to donate their journals to the project. More than fifty users agreed to have their journals included in the corpus, resulting in an untagged corpus of more than 2,000,000 words. The texts are written by speakers of Russian residing in the Russian Federation, the United States, Latvia, Israel, Australia, Estonia, and Belarus. The texts range from a sentence or two to several hundred words. We parsed the data using a suite of programs written in Perl, a programming language with very good text-handling capabilities, hand-tagged a small portion of the corpus with part-of-speech information, and began automatically tagging the remainder of the corpus.

3 Corpus Development and the Bazaar Model

Traditionally, corpora have been developed according to what is called the "cathedral" model in the software engineering community, where all members of a project adhere to a plan and are responsible for writing specific pieces of software or annotating specific data (Raymond 2001). They are developed almost exclusively within large research universities or national institutes of science and their licensing restricts use to academic or non-commercial settings.

The "bazaar" model is a fundamentally different way of running a project. Instead of following a specified plan and working only on an assigned section of the project, workers in the bazaar model focus their work on the functionality they need at the moment. Project managers act as moderators and assemble the sometimes-overlapping results into a coherent whole. This model is very effective for a number of reasons.

First of all, open-source programmers, who have more than a decade's experience with bazaar-style management in nearly every major software project from the Linux operating system down to smaller e-mail utilities like Fetchmail and SpamAssassin, have internalized a very important maxim: necessity is the mother of invention. Open-source projects succeed in developing robust, extensible software because there is a need or desire for such software and programmers flock to the project, eager to produce working code even if it means donating their time. This notion is reflected in popular open-source licenses like the GNU Public License and the BSD Artistic License, which allow anyone to modify code for their own purposes and even sell the results.[3] These licenses serve corporate and non-corporate commu-

[3]The GNU Public License was developed by the GNU Project, found at http://www.gnu.org/. The BSD Artistic License was created by the project at Berkeley that developed the Berkeley Standard Distribution, one of the first free Unices.

nities well, allowing corporations to package software and support for sale and permitting individuals, not-for-profits, and academic institutions to use the software for their own purposes.

Second, open-source programmers understand that "many eyeballs tame complexity" (Raymond 2001). Huge projects like the Mozilla web browser are kept optimized and free of bugs because millions of people use them each day and thousands of those users are willing to review code and submit patches or bug reports to project managers when a problem arises. While a cathedral-style project may take weeks or months to fix a reported bug and release a revision, depending on where such fixes fall in its overall plan, a user who reports a bug to an open-source project is often able to download and install a patched version of the software within hours.

Presently, corpus development projects in academia do not take into account the first maxim discussed above. Licenses that prohibit commercial use of corpus data remove any incentive for software companies to contribute time and money to development projects. The example of the open-source software community, where IBM and SGI are important partners, especially in the Linux kernel project, the core of the Linux operating system, software which forms the basis of the free Linux operating system, suggests this is a mistake. Furthermore, if linguists and computer scientists were presented with a few enticing applications of corpus data—an open-source morphological analyzer for use in the classroom or perhaps machine translation software, in the case of a two-language parallel corpus—the open-source software experience suggests they could be brought on board as annotators and programmers as well. However, neither strategy has been attempted thus far by any major corpus development effort.

The second insight about taming complexity can also be applied to a problem with traditional corpora: errors. Annotator errors are inevitable in any corpus, but Blaheta notes that this is especially a problem in the testing sections of corpora used for research. Algorithms that perform better than human annotators on a particular task are penalized because the human annotator's result is taken as the gold standard (Blaheta 2002). However, ethical considerations dictate that researchers not view testing data, as algorithms are evaluated by their performance on a few well-known corpora; a researcher who delves into the testing corpus to fix errors can never again say without reservation that his or her algorithms are not biased in favor of the test data (Magerman 1994). Blaheta suggests a solution to the problem, but it still does not permit anyone to modify the testing data. He also notes the problems that occur when researchers begin distributing their own changes to corpora. The bazaar model solves all of these problems: in a bazaar-style

corpus project, non-researchers are free to correct errors in the testing data, which they are free to use as part of their training data, and submit them to the project's maintainer for inclusion in the next release of the corpus. The ethical considerations raised by Magerman do not apply, and fixes are applied in an orderly manner and tied to specific releases of the corpus.

Applying a bazaar approach does result in new drawbacks, however. For instance, because the project cannot rely on dedicated paid labor to annotate data, a bazaar-model corpus project must attract a critical mass of volunteer annotators to ensure the long-term viability of the project. Also, the community-centric nature of such projects creates an increased danger of battles over features in the project, sometimes leading to schism.[4] Good project management and a clear sense of what the community needs from the project can go a long way to minimizing these problems.

4 Data Format

There are nearly as many data formats as there are corpora, so any budding corpus project must take care to select a format appropriate both to the project's purposes and the tools that will be used to interact with the corpus. Much of the code that interacts with the Penn Treebank, an important English-language corpus, is written in LISP, a language developed for "LISt Processing" in the 1950s, so the data are represented in the parenthetical "s-expression" format characteristic of LISP (Marcus et al. 1994). The Prague Dependency Treebank, on the other hand, is accessed primarily with software written in Perl and uses an SGML (Standard Generalized Markup Language) data format (Böhmová et al. 2003).[5]

For our corpus, we chose to use XML, the "eXtensible Markup Language." XML is a subset of SGML and is rapidly gaining acceptance in many different fields as a standard way of representing data. It uses beginning and ending tags around units of data; for instance, `<year>2004</year>` might represent the year 2004. In our corpus, sentences are delimited by `<s>` tags and words are delimited by `<w>` tags, as shown in Figure 1 on page 8.[6] Each tag can be assigned several attributes, such as the `id` attribute of `<w>`, which indicates

[4] See `http://www.xemacs.org/About/XEmacsVsGNUemacs.html` for an account of the split within the Emacs text editor project, which eventually led to the development of the XEmacs editor.

[5] SGML is the data format from which HTML, the code used to write Internet web pages, was derived.

[6] In this example, as in all examples throughout this thesis, the original Cyrillic text has been transliterated into the Latin alphabet.

a word's position in a sentence. The data are broken up into posts as they were originally published to the LiveJournal site. The date and time they were published is preserved, along with the title of the post.

```
<post>
 <year>2004</year>
 <month>01</month>
 <date>01</date>
 <apparent_time>14:21:00</apparent_time>
 <data>
  <s id="1">
   <w id="1">Ваня</w>
   <w id="2">пьет</w>
   <w id="3">чай</w>
   <w id="4">.</w>
  </s>
 </data>
 <title>чаепитие</title>
</post>
```

Figure 1: A sample post in the format of the corpus.

5 Tagset

A tagset is the collection of all parts of speech distinguished by a particular corpus. There are many different tagsets which are constructed using criteria that vary from corpus to corpus. The Penn Treebank, for example, distinguishes 48 parts of speech, including separate tags for dollar signs, commas, and the word 'to' (Marcus et al. 1994), while the Prague Dependency Treebank distinguishes only fourteen parts of speech (Hajič and Hladká 1998). Such tagsets also include varying amounts of morphological information, depending on the way the project goes about annotating its data.

Incompatible tagsets within and between languages are an acknowledged problem in the area of Slavic NLP (Przepiórkowski and Woliński 2003). However, most of the concerns raised by Przepiórkowski and Woliński—the mapping of duals in some languages to plurals and concepts of case and gender that do not match traditional tagsets, for instance—do not apply to Russian, which matches the categories in traditional tagsets rather well. Because

the tagsets developed by the Multext-East project (`http://nl.ijs.si/ME/`) have had some measure of success and have been adopted by a number of projects in other Slavic languages, we have chosen to use a subset of the Multext-East–derived Tübingen tagset to annotate the corpus, which is shown in Figure 2 on page 10. However, where the Tübingen tagset includes a great deal of morphological information, our tagset is fairly basic and tries mainly to distinguish major classes of words with significantly different distributions, leaving out most morphological information at the part-of-speech level.

For example, because we have based our decisions about the tagset on distribution, long- and short-form adjectives are given distinct tags, but possessive adjectives like мамин are tagged as adjectives and are not given a class of their own. Section 8 explains how the tagger can determine they are adjectives even though such possessives can be formed from almost any proper name. Also, while the Tübingen tagset identifies words like можно and надо as "residual," we tag them as modals along with modal verbs like хотеть and иметь. See Appendix B for more notes on annotating the corpus.

Finally, although almost all natural language corpora contain a "foreign word" part-of-speech tag, we have chosen to approach the problem of foreign words in corpus data differently. Instead of assigning the FW (foreign word) tag to every foreign word in the corpus, we have chosen to assign them an `fw` attribute. In addition to this attribute, they will be assigned a normal part-of-speech tag just like other words in the corpus.

However, this attribute is reserved almost exclusively for words written in alphabets other than Cyrillic. Although some may balk at the annotation of words like сканер as Russian words, in many cases words of foreign origin, especially ones dealing with technology, are being assimilated into the language and are no longer truly foreign. For example, the acronym SMS (Short Message Service, used for sending short text messages on cellphones) has entered Russian and appears in one post as СМСок—it has been so thoroughly assimilated into the language by some speakers that it receives the diminutive suffix -ок there.

6 Sentence Boundary Detection and Tokenization

We used a direct-model sentence boundary detection algorithm based on Manning and Schütze's simple heuristic (Walker et al. 2001; Manning and Schütze 1999), then corrected the resulting data by hand. A sentence boundary was marked everywhere a period, question mark, or exclamation mark

Tag	Part of Speech	Example
A	Adjective	старый
ASF	Short-form Adjective	нездоров
ADV	Adverb	быстро
CCN	Subordinating/Complementizing Conjunction	хотя
CCNJ	Coordinating Conjunction	и
CD	Cardinal Number	один
CMPJ	Comparative Adjective	красивее
CMPV	Comparative Adverb	легче
DT	Determiner	каждый
EMOT	Emoticon	:))
GER	Gerund	думая
INT	Interjection	ой
MOD	Modal	нельзя
N	Noun	дом
NA	Animate Noun	муж
NEG	Negation	не
NP	Proper Noun	Москва
NPA	Animate Proper Noun	Иван
PC	Particle	же
PER	Period	.
PRA	Active Participle	любящий
PREP	Preposition	в
PROA	Possessive Pronoun	мой
PRON	Personal Pronoun	я
PRP	Passive Participle	сделанный
PSTP	Postposition	ради
PUN	Punctuation	,
QU	Degree or Question Word	кто
QUOT	Quotation Mark	"
SUPJ	Superlative Adjective	красивейший
SUPV	Superlative Adverb	наиболее
SYM	Symbol	@
V	Verb	смотрю
VINF	Infinitive	смотреть’

Figure 2: Our Tagset for Russian.

preceded a word beginning with a capital letter, as well as following emoticons (textual representations of facial expression like `:)` and `:o` popular in chatrooms and web forums) preceding a capital letter. However, no boundary was placed if the word preceding a particular period was in a dictionary of abbreviations that do not typically occur sentence-finally (e.g. св.). This simple algorithm achieved an accuracy of 96% on the data annotated over the course of the project.

Because we used XML as the basic format for the corpus, we were able to place sentence boundaries around sentences embedded in other sentences, something corpora with simpler representations do not permit—see examples from the Prague Dependency Treebank in Böhmová et al. (2003). For the purposes of the corpus, we define an embedded sentence as a sequence of words that can stand as a sentence on its own and is separated from the rest of the sentence by punctuation "stronger" than a comma (usually a semicolon, a dash, or parentheses). Text separated by such "stronger" punctuation that does not consitution a separate sentence simply remains a part of its parent sentence without further annotation. This allows an annotator to deal easily with the problems posed by long quotations and sentences inside parentheses.

Figure 3 on page 12 illustrates how we handled sentences with other sentences embedded in them. Notice how the king's first question to Beren is a sentence embedded in the sentence that describes the asking of the question, which is in turn embedded in the top-level sentence. Throughout the corpus, `<s>` marks the beginning of a sentence, and `</s>` marks the end of one.

Unfortunately, one disadvantage of annotating embedded sentences is that such sentences are very difficult for the simple sentence boundary detector described above to identify. To correctly identify such sentences, the detector would need to take into account part-of-speech and syntactic information, which is not available at this stage of annotation. For the same reason, sentence fragments are impossible for the detector to identify without having more information about the text. These must be identified and annotated by hand. If a user does not like the annotations for embedded sentences or is using software that cannot handle them correctly, they can be stripped out easily.

Tokenization, the process of separating a sentence into its most basic units, is a fairly simple process by comparison. Tokens are very similar to what we think of as words, but some things we do not typically consider words are considered tokens. Many punctuation marks are considered tokens, though they are not words, and some words are composed of more than one token. For instance, in the word наконец-то, наконец and -то are considered

(1) <s> *Местами попадались просто потрясающие фразы*
Places turned-up simply staggering phrases
(<s> цитата приводится по памяти </s>): “<s>—<s>
(quotation cited by memory):
У тебя есть совесть, Берен? </s> — спросил
“By you is conscience, Beren? — asked
эльфийский король. </s><s> Берен перегнулся через
Elvish king. Beren leaned-over through
перила. </s> — <s> Да, государь, у меня есть
railing. — Yes, sir, by me is
совесть, и я об этом много думал. </s>”</s>
conscience, and I about this much thought.”

‘In a few places I found phrases that were simply staggering (I’m posting these quotes from memory): “— Do you have a conscience, Beren? — asked the Elvish king. Beren leaned over the railing. — Yes, sir, I have a conscience, and I’ve thought about this a lot.”

Figure 3: A complex sentence with several embedded sentences.

separate tokens.

In our algorithm, any sequence of letters and symbols surrounded by spaces is tentatively considered a token. Any sequence ending in a period followed by a sentence boundary is compared to a list of abbreviations that can appear sentence-finally; if it does not match the list, the period is separated and made its own token. Finally, most other punctuation marks are separated from the words they attach to, including commas, quotation marks, and parentheses. The only exceptions are emoticons, which are each considered a single token, and word-medial apostrophes, which are often used when inflecting foreign words and names.

7 Word-level Annotation

Even in a language with “free” word order like Russian, words are placed in sentences according to fairly specific rules. For example, verbs never follow prepositions, because prepositions take noun phrases as arguments. In part-

of-speech tagging, software observes the frequency with which certain parts of speech follow other parts of speech and the frequency with which a certain part of speech is associated with a specific word. For example, a noun might follow a preposition 15% of the time, and сегодня might be a noun 40% of the time. The software then uses these known probabilities to guess at the parts of speech of words in texts it has not seen before. It is useful to consider all these probabilities because many words take on different parts of speech depending on context, though such ambiguity is more infrequent than in English. For example, что can be used as either a question word or a subordinating conjunction; which part of speech should be assigned to it depends on context.

So for each word in a sentence, a tagger determines how likely it is for a given part of speech to follow the previous part of speech *and* to be associated with the particular word in question. This is illustrated in Figure 4 on page 14, where s_{t-1} is the previous part of speech, s_t is a possible part of speech for the current word, and w_t is the current word. Even if a particular combination of parts of speech is highly likely (adjective followed by noun, for instance), if the combination of part of speech and observed word is very low (noun and the word в, for example), the tagger will likely choose a less-likely pairing of parts of speech which maximizes the likelihood of the entire combination. Likewise, if a particular combination of part of speech and observed word is very likely (verb and еду) but the resulting pairing with the previous part of speech is unlikely (preposition followed by verb), the tagger will choose a less likely tag for the word that again maximizes the entire combination. This is done for each word in the sentence, resulting in a single most likely combination of all tags and words.

One possible way of doing this is to calculate the probability of every possible combination of tags. Unfortunately, the amount of work required by this method grows exponentially as sentences grow, making it impractical for all but the shortest sentences. Instead, most supervised[7] part-of-speech taggers use the Viterbi method, which allows a tagger to eliminate all but n combinations of tags, where n is the number of tags in the tagset (Viterbi 1967). This method has been mathematically proven to preserve the most likely combination of tags.

[7]Supervised taggers differ from unsupervised taggers in that they have been trained on a previously annotated body of data.

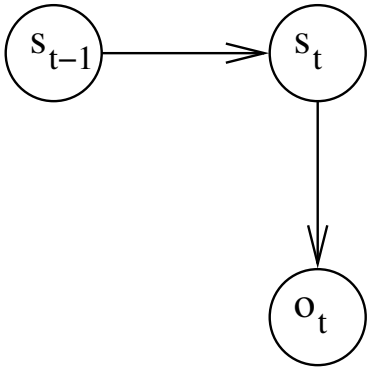

Figure 4: Old dependency graph.

8 Strategies for Dealing with Sparse Data

Most part-of-speech taggers require a large amount of training data to estimate the probabilities needed for this process. For instance, in its early stages the Penn Treebank Project benefited from the PARTS tagger (Marcus et al. 1994), which was trained on the Brown Corpus, approximately one million tokens of hand-tagged text (Church 1988). The NEGRA treebank of German found they could begin automatically assigning tags based on only a few thousand tokens of hand-tagged text, but they note this depends on the language in question (Brants et al. 2003). For instance, a language that has relatively few words with ambiguous parts of speech, like Russian or German, is likely to require a smaller training set to begin automatic tagging than a language like English, with many words whose part of speech cannot be easily determined.

Working with a language as morphologically diverse as Russian, we were already concerned about sparse data problems, where specific words are observed only a few times and provide misleading information. Without doing some kind of morphological analysis—which would require a great deal of training data and some part-of-speech information—the various forms of one word (different inflections of a single word, for instance) are understood by the software as different words whose part of speech is not related to the part of speech of other forms of that word. This means that although the software may have observed the word 'voditel'' many times in training, if it

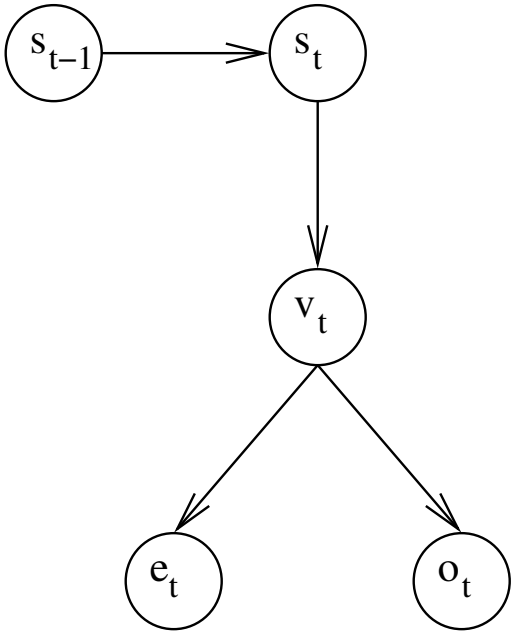

Figure 5: New dependency graph.

encounters the dative form 'voditelju', it will be just as confused as if it had encountered an entirely new word. These concerns were exacerbated when it became clear there would only be enough time for us to hand-tag about 2000 tokens before entering the automatic stage of tagging. Because of this, we decided to adopt a slightly modified algorithm to decrease reliance on forms observed only a few times in training.

Usually, the predicted state (in this case, the part-of-speech tag) is determined using the previous state and the current word, as shown in Figure 4. The modified version, shown in Figure 5, makes use of the suffixes of words that have not been observed more than five times. If the word in question *has* been observed at least five times, the tagger relies on the probability o_t of observing that form given the part of speech and the probability that the word is associated with that part of speech. If the word has *not* been seen five times, the tagger relies on the probability e_t that a suffix of a certain length is associated with the part of speech. Since the last few letters of Russian words are often rather indicative of their part of speech, this min-

	1500 words	2300 word
Naïve	58.1%	60.1%
2-letter	72.6%	76.6%
Principled	74.0%	78.3%

Figure 6: Tagger accuracy results.

imizes problems associated with word forms the tagger has seen only a few times. This is one of several common strategies for dealing with sparse data (Manning and Schütze 1999).

At first we simply dealt with the last two letters of each word, but in the latest version of the tagger, the last three letters of the word are usually considered. The tagger considers the last four letters if the word ends in the reflexive suffixes -ся or -сь. If the word ends in -вши or contains the infix -ающ-, the tagger considers the last five letters, and if a reflexive suffix -ся or -сь is present in addition to endings typical of a participle, the last six letters are considered.

These modifications to the algorithm significantly improved the accuracy of the tagger. Figure 6 shows how the tagger performed when trained on training sets of 1500 and 2300 words and tested on a small testing corpus of 500 words. Each model was tested using the naïve tagger, the two-letter model, and the more principled suffix model. In addition to testing the tagger on Internet-register texts, we also had the software tag the Chekhov short story "The Bridegroom," which belongs to a higher register of Russian. Here the tagger, trained on 2300 words of Internet text, performed admirably, achieving an accuracy of 75.4%. In the future, we hope to add texts from other registers, such as literary Russian, to the corpus; this should increase the tagger's accuracy on such texts.

9 Reducing Human Intervention

The experience of the NEGRA corpus of German also suggests accuracy can be increased and human annotation time significantly reduced by red-flagging words that are deemed unreliable (Brants et al. 2003). They define a threshold θ where

Best Tag (Prob.)	Second Best Tag (Prob.)	θ
N (99%)	V (0.01%)	9900
N (99%)	V (1%)	99

Figure 7: Calculating θ.

$$\theta = \frac{P(t_{best})}{P(t_{second})},$$

$P(t_{best})$ is the probability of the best tag, and $P(t_{second})$ is the probability of the second-best tag. If θ is below this threshold, the tag assignment is assumed to be unreliable. In Figure 7, we see that N (noun) is probably a safe assignment in the first case, but a human annotator's intervention is probably required in the second.

Using $\theta = 100$, the NEGRA project found that 84% of all words in the corpus were reliable and were tagged by the computer with an accuracy of 99.3%. The unreliable words, the remaining 16% of the corpus, were only tagged with an accuracy of 82.7%. By focusing only on unreliable assignments, the project was able to achieve 99% accuracy by correcting only 16% of the corpus. We have not been able to test these insights on the Russian corpus yet, but there is no reason to expect they will not be as useful for Russian as they are for German.

The results are encouraging—even with a training set as small as 2300 words, we can get the computer to do nearly four-fifths of the work of tagging for us. However, it is difficult to tell how the tagger rates against state-of-the-art part-of-speech tagging software, because no other corpus development projects have released their results at this early level of annotation. More data are necessary to see what kinds of innovations could make the tagger even more effective than it already is.

10 Processing of a Sample Post

First, Aleksandr Nikolayev's `ljsm` script (available at `http://www.offtopia.net/~ati/ljsm/`) is used to download the post and strip out a large amount of the extraneous HTML. The `clean_data` script removes the remainder of

the HTML and re-writes the post in the XML format used for the corpus. The resulting file contains a small amount of data about the post and the text of the post with no formatting:

```
<post>
 <year>2004</year>
 <month>01</month>
 <date>01</date>
 <apparent_time>14:21:00</apparent_time>
 <data>
О. Григорий сказал, "Ваня пьет чай."
 </data>
</post>
```

Next, the `sentencify` program assigns sentence boundaries, which are then corrected by hand. The sentence Ваня пьет чай is marked as an embedded sentence.

```
<post>
 <year>2004</year>
 <month>01</month>
 <date>01</date>
 <apparent_time>14:21:00</apparent_time>
 <data>
  <s>О. Григорий сказал, "
   <s>Ваня пьет чай</s>
  ".
  </s>
 </data>
</post>
```

The resulting sentences are automatically tokenized as described in Section 6. Here we see that the abbreviation О. (for отец), which does not usually occur sentence-finally, keeps its period as part of the same token.

```
<post>
 <year>2004</year>
 <month>01</month>
 <date>01</date>
 <apparent_time>14:21:00</apparent_time>
 <data>
```

```
  <s>
   <w>О.</w>
   <w>Григорий</w>
   <w>сказал</w>
   <w>,</w>
   <w>"</w>
   <s>
    <w>Ваня</w>
    <w>пьет</w>
    <w>чай</w>
   </s>
   <w>"</w>
   <w>.</w>
  </s>
 </data>
</post>
```

The `mytagger` program assigns tentative part-of-speech tags using data compiled by the trainer program running on the training corpus. Finally, a human annotator corrects the part of speech assignments and includes the corrected post in the corpus. The final result, shown below, is integrated into the training set to improve the tagger's model for assigning part-of-speech tags.

```
<post>
 <year>2004</year>
 <month>01</month>
 <date>01</date>
 <apparent_time>14:21:00</apparent_time>
 <data>
  <s>
   <w pos="NA">О.</w>
   <w pos="NPA">Григорий</w>
   <w pos="V">сказал</w>
   <w pos="PUN">,</w>
   <w pos="QUOT">"</w>
   <s>
    <w pos="NPA">Ваня</w>
    <w pos=">пьет</w>
    <w pos="N">чай</w>
```

```
   </s>
   <w pos="QUOT">"</w>
   <w pos="PER">.</w>
  </s>
 </data>
</post>
```

The `check_tags` script can be used to check the computer's accuracy against corrected files. All the software mentioned above, with the exception of `ljsm`, is available on the web at `http://www.polyglut.net/corpus/`. Appendix A contains a much longer annotated post.

11 Conclusions

Unfortunately, it is still too early to tell whether corpus development is similar enough to software development for the bazaar model to work. However, things do look promising. In working on the first stages of this corpus, we have encountered a number of Russian software developers and amateur linguists who are excited about the prospect of having access to the data. In particular, many people would like access to an open-source stemmer (a program that finds the basic form of a word from one of its inflected forms) and morphological analyzer like the well-known Lingvo software (see `http://lingvo.yandex.ru`). There is reason to hope the considerable programming resources of the Russian open-source community can help drive development of such tools and the data necessary to train them.

Once all the necessary software was written and texts were obtained, it took one annotator about twenty hours to develop a training set of 2300 words and train a tagger capable of assigning part-of-speech tags with an accuracy of nearly 80%. Because the programs developed for this project are all available under the GNU Public License, which allows anyone to use and modify them for almost any purpose, future corpus projects can use the software to quickly develop preliminary training sets and automatically annotate data.[8] Likewise, the data are licensed under a Creative Commons license that allows anyone to modify and use the text in many different ways, hopefully opening the door to creative applications of the corpus.[9]

[8] The GNU Public License is available at `http://www.gnu.org/licenses/gpl.txt`.

[9] The Creative Commons license that applies to the corpus data is available at `http://creativecommons.org/licenses/by-sa/2.0/legalcode`. For more information about the Creative Commons project, see `http://creativecommons.org/`.

At this stage, experienced annotators should be able to add new text to the corpus at a rate of about 450 words per hour: sentence-level annotation at 18,000 words per hour, tokenization at 10,000 words per hour, and part-of-speech annotation at 500 words per hour. The Penn treebank found that when the automatic tagger performed with an accuracy of 95%, its experienced annotators could correct part-of-speech tags at a rate of nearly 3000 words per hour (Marcus et al. 1994).

In this thesis, we have presented several strategies for making the task of tagging Russian texts faster and more accurate. We have also suggested a new model for corpus development that includes commercial and amateur users and spreads the financial and labor burdens among more people. If the insights gained in bazaar-model software development can be applied to corpus development, the barriers to developing corpora will be much lower, and we may see the development of freely-available corpora of other under-represented languages in the near future.

12 Further Work

Now that we have a small training corpus and a tagger capable of assigning parts of speech with reasonably accuracy, the next step is to complete tagging of a large corpus, perhaps 1,000,000 tokens. With this information available, the corpus can be annotated with morphological information, which will lead to the first serious application of the data, writing an open-source morphological analyzer to mirror some of the functionality available with the Lingvo software. Further applications, such as building stochastic grammars and retrieving discourse information from written texts, will require syntax-level annotation and function tags (Collins 1999; Blaheta 2003).

13 Acknowledgements

I would like to thank my advisor, Charles Mills, for his support during my time at Knox. His enthusiasm about Russian and Slavic linguistics got me interested in how we understand language in the first place.

I would also like to thank the other members of my committee, Don Blaheta and Gerald Krumbholz, my outside examiner, Steven Clancy, and everyone else who read drafts of my thesis and gave me feedback.

I am especially grateful to the Department of Computer Science at Brown University and the National Science Foundation for giving me the opportunity to study corpus linguistics and NLP with Eugene Charniak and others

in the Brown Laboratory for Linguistic Information Processing during the summer of 2003. That assistantship gave me the freedom to explore the Prague Dependency Treebank in depth, to read a great deal about the work being done in NLP, and to study basic concepts in corpus linguistics in the company of some very bright people.

I owe a great deal to Julia Aleksandrovna Husen of the Illinois Mathematics and Science Academy and Vasily Gregoryevich Fiedorow of Knox College, who guided me in my development as a speaker of Russian and reader of Russian literature. Without them, I never would have experienced Tolstoy, Gogol, Pushkin, and Dostoevsky the way they were meant to be experienced. I would also like to thank Genevra Gerhart for her encouragement in all matters Slavic over the two years I have corresponded with her.

Finally, I would like thank Dean Stephen Bailey for his support and sage advice during my four years at Knox, the brothers of the Delta Theta chapter of Sigma Nu for teaching me some of the most important lessons of my college years, my family for sticking with me no matter what I decided to do or where I decided to go, and my fiancée, Sue Massey, for her love and understanding. Without them, this thesis would not have been possible.

Appendix A: Sample Post

What follows is a sample post from the corpus in its final state.

```
<post>
 <year>2002</year>
 <month>04</month>
 <date>23</date>
 <apparent_time>11:51:00</apparent_time>
 <data>

  <s id="1">
   <w id="1" pos="V">Посмотрели</w>
   <w id="2" pos="N">квартиру</w>
   <w id="3" pos="ADV">наконец</w>
   <w id="4" pos="PER">.</w>
  </s>

  <s id="2">
   <w id="1" pos="ADV">Вроде</w>
```

```
 <w id="2" pos="PC">бы</w>
 <w id="3" pos="DT">всё</w>
 <w id="4" pos="ADV">более-менее</w>
 <w id="5" pos="PER">.</w>
</s>

<s id="3">
 <w id="1" pos="PREP">В</w>
 <w id="2" pos="N">воображении</w>
 <w id="3" pos="N">квартира</w>
 <w id="4" pos="V">казалась</w>
 <w id="5" pos="CMPJ">больше</w>
 <w id="6" pos="PER">.</w>
</s>

<s id="4">
 <w id="1" pos="N">Вода</w>
 <w id="2" pos="PREP">по</w>
 <w id="3" pos="N">стояку</w>
 <w id="4" pos="V">стекает</w>
 <w id="5" pos="ADV">сверху</w>
 <w id="6" pos="ADV">откуда-то</w>
 <w id="7" pos="PER">.</w>
</s>

<s id="5">
 <w id="1" pos="CCNJ">Но</w>
 <w id="2" pos="DT">это</w>
 <w id="3" pos="N">дело</w>
 <w id="4" pos="A">такое</w>
 <w id="5" pos="PER">.</w>
</s>

<s id="6">
 <w id="1" pos="ADV">Правда</w>
 <w id="2" pos="PUN">,</w>
 <w id="3" pos="V">выяснилось</w>
 <w id="4" pos="PUN">,</w>
 <w id="5" pos="CCN">что</w>
 <w id="6" pos="ADV">окончательно</w>
```

```
   <w id="7" pos="PRON">нам</w>
   <w id="8" pos="N">ключ</w>
   <w id="9" pos="V">отдадут</w>
   <w id="10" pos="ADV">только</w>
   <w id="11" pos="PREP">после</w>
   <w id="12" pos="N">праздников</w>
   <w id="13" pos="PUN">-</w>
   <s id="7" frag="1">
    <w id="1" pos="CCNJ">а</w>
    <w id="2" pos="CCNJ">поскольку</w>
    <w id="3" pos="V">вернёмся</w>
    <w id="4" pos="PRON">мы</w>
    <w id="5" pos="ADV">только</w>
    <w id="6" pos="A">20-го</w>
    <w id="7" pos="PUN">,</w>
    <w id="8" pos="DT">то</w>
    <w id="9" pos="PER">.</w>
   </s>
  </s>
 </data>
 <title></title>
</post>
```

Appendix B: Some Notes on Annotation

When annotating or correcting the corpus by hand, it is not always clear what tag should be assigned to a given word. The following are the guidelines we have developed over the course of the project.

- Although its primary use is as a conjunction, и is frequently used to add emphasis in a sentence. For example, in the sentence, "Упаковали полученный tiff в zip и так и переслали по мейлу," the second и is not acting as a conjunction. We tag it PC (particle).

- Some participles are used so frequently that they can seem like adjectives, and some constructions that are technically declined nouns, like бегом and порой, are usually used as adverbs. We deal with such words on a case-by-case basis and have begun compiling a list of problematic words and constructions, but in the case of such a construction

that has not been encountered before in tagging, Lingvo can be a useful guide.

- When encountering the construction было + past perfective verb to indicate an unrealized desire to do something, было is tagged PC (particle).

- Short-form participles (начитан, for example) are tagged ASF (short-form adjectives) for the time being. It may be useful to split the two categories when sparse data is a less serious concern.

References

Abeillé, Anne (Ed.). *Treebanks: Building and Using Parsed Corpora.* Kluwer Academic Publishers, 2003.

Allen, Cynthia L. "The Early English '*his* Genitives' from a Germanic Perspective." *Proceedings of the 2002 Conference of the Australian Linguistic Society.* Also available at `http://www.arts.unsw.edu.au/als2002/proceedings/Allen.pdf`, 2002.

Blaheta, Don. "Handling noisy training and testing data." *Proceedings of the 7th conference on Empirical Methods in Natural Language Processing.* July 2002, 111–116.

Blaheta, Don. *Function Tagging.* Ph.D. dissertation, Brown University, 2003.

Böhmová, Alena, Jan Hajič, Eva Hajičová, and Barbora Hladká. "The Prague Dependency Treebank: A Three-Level Annotation Scenario." *Treebanks: Building and Using Parsed Corpora.* Ed. Anne Abeillé. Kluwer Academic Publishers, 2003. 103–127.

Brants, Thorsten, Wojciech Skut, and Hans Uszkoreit. "Syntactic Annotation of a German Newspaper Corpus." *Treebanks: Building and Using Parsed Corpora.* Ed. Anne Abeillé. Kluwer Academic Publishers, 2003. 73–87.

Church, Kenneth W. "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted text." *Proceedings of the Second Conference on Applied Natural Language Processing.* Also available at `http://acl.ldc.upenn.edu/A/A88/A88-1019.pdf`, 1988.

Collins, Michael. *Head-Driven Statistical Models for Natural Language Parsing.* Ph.D. dissertation, University of Pennsylvania,

1999. Also available at http://www.ai.mit.edu/people/mcollins/papers/thesis.ps.

Hajič, Jan and Barbora Hladká. "Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset." *Proceedings of COLING-ACL Conference*. 1998, 483–490.

Hana, Jiři and Hana Hanová. Manual for Morphological Annotation. Technical Report TR-2002-14, Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University, 2002. Also available at http://shadow.ms.mff.cuni.cz/pdt/Corpora/PDT_1.0/References/mman.html.

Johansson, Stig and Anna-Brita Stenström (Eds.). *English computer corpora: selected papers and research guide*. Mouton de Gruyter, 1991.

Kurohashi, Sadao. "Building a Japanese Parsed Corpus While Improving the Parsing System." *Treebanks: Building and Using Parsed Corpora*. Ed. Anne Abeillé. Kluwer Academic Publishers, 2003. 249–260.

Magerman, David M. *Natural Language Processing as Statistical Pattern Recognition*. Ph.D. dissertation, Stanford University, 1994. Also available at http://arxiv.org/pdf/cmp-lg/9405009.

Manning, Christopher D. and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. "Building a Large Annotated Corpus of English: The Penn Treebank." *Computational Linguistics* 19 (1994): 313–330.

McEnery, Tony and Andrew Wilson. *Corpus Linguistics*. Edinburgh University Press, 2001.

Möllering, Martina. "Teaching German Modal Particles: A Corpus-Based Approach." *Language Learning & Technology* 5 (2001): 130–151.

Przepiórkowski, Adam and Marcin Woliński. "A Flexemic Tagset for Polish." *Proceedings of Morphological Processing of Slavic Languages, EACL 2003*. Also available at http://dach.ipipan.waw.pl/~adamp/Papers/2003-eacl-ws12/ws12.pdf, 2003.

Raymond, Eric S. *The Cathedral and the Bazaar*. O'Reilly & Associates. Also available at http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/, 2001. 19–63.

Sampson, Geoffrey. "Thoughts on Two Decades of Drawing Trees." *Treebanks: Building and Using Parsed Corpora*. Ed. Anne Abeillé. Kluwer

Academic Publishers, 2003. 23–41.

Viterbi, A.J. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm." *IEEE Transactions on Information Theory* IT-13 (1967): 1260–69.

Walker, Daniel J., David E. Clements, Maki Darwin, and Jan W. Amtrup. "Sentence Boundary Detection: A Comparison of Paradigms for Improving MT Quality." *Proceedings of MT Summit VIII*. Also available at `http://www.eamt.org/summitVIII/papers/walker.pdf`, 2001.