Chapter 5  
The Devil is in the Details:  
Digital Analysis of Gor’kii and the Christian Tradition[[1]](#footnote-2)

In distant antiquity when humans began to reason, they reasoned technologically, which is to say exclusively relying on and only on one’s own labor experience. Technology is the logic of facts created by people’s labor, and ideology is the logic of ideas, which is to say the logic of thoughts extracted from facts, from thoughts that dictate the paths, devices, and forms of the creation of new facts.  
Maksim Gor’kii, “On Formalism” (1936)[[2]](#footnote-3)

Introduction

Despite Orthodox Christian culture’s obvious, profound impact on Gor’kii’s worldview and creative imagination, the author categorically denied any such religious influence. Having demonstrated some of the qualitative evidence of this phenomenon, I believe there is quantitative proof to corroborate and expand past conclusions. This chapter discusses “forensic” Digital Humanities (DH) to foreground an empirical search for the truth of Gor’kii’s well-known secret. The following describes how computational thinking and DH methods can strengthen the evidentiary basis for asserting the substantial role of religious literature in Gor’kii’s work. Using Natural Language Processing (NLP) tools in Python, I establish a definitive connection between some of Gor’kii’s major pre-revolutionary texts and the Christian literary tradition, especially the Russian Synodal Bible [*Synodal’nyi perevod Biblii*] (1885). These digital instruments allow me to show like never before the myriad parallels between his works, which the Soviets would weaponize against religion, and Orthodoxy in nearly all but name.

A machine-readable Biblical and literary corpus constitutes the primary product of the technical labor described. ALGO From plain-text files I created a fully indexed, searchable database of digital texts. Each text has its own features, but more importantly, the database presents cross-text (shared) linguistic and semantic data. I extracted this data using a proprietary program created in the programming language Python and organized it in a mySQL relative database. On the front end, I created a user interface for querying the database using a “vanilla”—i.e. from scratch—combination of PHP, HTML, CSS, and JavaScript. The UI displays custom XML documents of the texts that are annotated or “marked-up” to allow for users’ interaction with the intertextual elements, Gor’kii’s secular transpositions, discussed elsewhere in the project. Visualized links show side-by-side how Gor’kii took from the Bible to create his own “revolutionary gospel.” These are my methods and preliminary findings.

*Background*

Textual DH has a rich history and is currently taking exciting leaps thanks to recent technological advances. Cardstock punch cards and manual term counting of the past have evolved with general technological advancement alongside theories that push DH beyond simply numbering the vocabulary of a corpus, which is any collection of texts. Milestone works such as *An Introduction to Information Retrieval* by Chistopher Manning et al. (2009), *Mining Text Data* by Charu Aggarwal and Chengxiang Zhai (2012), and *Speech and Language Processing* by Daniel Jurafsky and Jacob Martin (2018) continue to serve as invaluable primers for computational approaches to texts and their meanings. Many instructional texts in the Russian NLP sphere are translated from English. A welcome Russian source is *Automatic Processing of Natural-Language Text and Data Analysis* [*Avtomatichestkaia obrabotka tekstov na estestvennom iazyke i analiz dannykh*] (2017) by E. I. Bol’shakova et al. which provides a similar introduction to computer-aided approaches for Russian-language NLP.

Although Russian-language NLP is developing more slowly other major world languages, some existing tools have proven useful. As a result of NLP’s demand for expensive computational resources, work is often concentrated in Western countries where such demands can be met, and the English language is dominant. Thus, while the statistical principles underlying NLP methods are language independent, the quantity of pre-packaged NLP tools drops off precipitously as one leaves the languages of North America and Western Europe. Russian NLP is pushed forward largely by researchers at Yandex, who want to keep up with advancements made at their American counterpart, Meta (Facebook), and individual researchers.[[3]](#footnote-4) Two such independently developed projects are DeepPavlov and Natasha.[[4]](#footnote-5) DeepPavlov and Natasha both provide the basic tools for researchers to extract data from Russian-language texts. These programs are designed to give Russian speakers fine control over information gathering from texts.

The morphological and grammatical complexity of the Russian language has been a factor in its slow development, but it also presents exciting challenges to tackle. A highly inflected morphology and flexible syntax produce numerous semantic layers across a sentence or even single word. That is to say that between English and Russian utterances conveying the same idea for all intents and purposes, the Russian sentence will contain more information about the idea—and contain less “noise,” or meaningless information—than the English. Grammatical rules dictate how to identify a word’s part of speech and then dissect its attributes. Understanding this system offers a bevy of potential connections between texts and their parts in addition to the words’ practical literary meaning. Although both DeepPavlov and Natasha allow researchers to access these linguistic features, Natasha’s more economic approach to near identical results was sufficient reason to choose it as one of my foundational tools. With this information, I could probe my corpus, the group of texts with which I am working, with questions that Gor’kii would not answer.

*Methods*

In search of common language use between literary works and the Bible, I employ a “sliding token-window” comparison technique at a granular level. A “token” is a particular instance of a word within a text. While a word potentially can be found in multiple places across a text, a token is by definition unique based on its location. A token window is a contiguous grouping of tokens. Each text is a series of tokens that can be viewed through a sliding, or moving, window of *n* tokens at a time. For example, a three-token window (i.e., three consecutive words) could be comprised of the first through third, second through fourth, third through fifth tokens, and so on. Any number of consecutive tokens can be arbitrary grouped to define a “string.” The number of possible groups of *n* words in a set of *s* is . A ten-token string can be sliced into eight possible three-token windows. When measuring the similarity of substrings between two or more extended strings, which is how we may abstractly characterize transpositions on a multi-token level, the sliding token window allows for a high degree of accuracy in locating and quantifying borrowed language. For this reason, I devised a computational mechanism to compare the literary works and Synodal Bible by dividing each text into “*n*-token” windows (i.e., word strings of a given “n” length) and counting the coincidence of tokens between each window of the literary work with each window of the Bible.

The token-window method is facilitated by first determining the “lemma” of each token, which is the true object of comparison. The lemma is the standardized form of a token without additional morphemes. For example, the English words “[she] plays,” “[he is] playing,” and “[they] played” are all the verb “play.” Thus, comparing only lemmas allows for the control of context-dependent information that represents noise for the purpose of counting shared tokens. This is particularly important in Russian because of the high variance found in inflected forms of nouns and adjectives as well as conjugated verbs. Whereas English has a handful of verbal inflections and even fewer of those of nouns and adjectives, Russian features at least six verbal and twelve nominal and adjectival inflection forms each. The lemma reduces the dozens of forms to a single word and thus greatly simplifies the comparison process. Considering the lemma of a word is preferable to its stem, which is the minimal part of a word conveying solely its semantic significance, because the lemma preserves the syntagmatic and pragmatic qualities of a token. In other words, stemming would conflate the verb "starts," the adjective "starting," and the nouns “start” and “starter.” Lemmatization, the process of determining token’s lemma, is a standard and important function of NLP software packages like Natasha. They accomplish this goal much like a human would: examining the token, considering its part of speech and other features, making appropriate morphological modifications, and comparing the result to a list of known lemmas of the language. Natasha is admittedly less capable of handling edge cases and exceptions compared to a scholar. However, though a person may be able to eventually produce a lemma list with 100% accuracy, Natasha can create a list of all the lemmas of the Bible’s tokens in under five minutes with greater than 99% accuracy. At the scale of data under examination, that <1% error rate is an acceptable loss to accomplish what would take a person thousands of hours of labor to accomplish.

The help Natasha and other technological tools provide is invaluable because of the sheer number of tokens involved in the task at hand, which also left me with a question of exactly what tokens my code should consider when comparing texts. However, not all tokens are equally meaningful. After considering other transpositions, I created a filter to remove a portion of the tokens from each text in order to more efficiently locate the overlapping tokens that are rich in content and lack noise to the extent possible. To that end, I leveraged Natasha’s part-of-speech (POS) tagger and distilled the token lists based on their label. The tokens permitted to pass through the filter for comparison were those tagged as nouns, including proper nouns, adjectives, and verbs. These tokens in theory represent the most meaningful and least noisy of POS, as opposed to those excluded: adverbs, exclamations, numbers, conjunctions, determiners, and others.[[5]](#footnote-6) From the total Bible token count, approximately 51.11% (345,090 of 675,068) tokens, to be exact were preserved. A slightly greater proportion of 54.46% (46,836 of 86,009) of Mother’s tokens passed through the filter. Whittling down these token sets has provided me with a more focused token set from each text in hopes of isolating windows demonstrating Gor’kii’s use of Biblical language. At the same time, the numbers are nothing to balk at; with several billion comparisons to make, the possibilities are nearly limitless.

Getting to the actual counting of windows’ overlapping tokens, it is important to explain what numbers are available. Each text is divided into n-token windows, which are then each compared against each other. A ten-token window methodology renders 373,398 windows from 345,090 total tokens from the Bible and 54,729 windows from 46,836 tokens from *Mother*. This example provides for 20.435 billion possible comparisons. Each window is assigned an identification number, and upon comparison with another window, a key-value pair is generated, in which the unique window-window identification is the key, and the similarity is the value. The key-value pair is a convention of Python dictionaries, which dictates that an unordered list of data can contain a unique set of keys, each of which corresponds to a value that is not necessarily unique. An additional consideration is the comparison of Bible words in their established sequence. The differences between the methods’ results are drastic. On the token-window level, the question about how to measure similarity is straightforward. The sequence-dependent method counts matches if the position of a given token in Gor’kii’s works matches the same position in the window from the Bible.

Human intervention is nevertheless necessary to sift through the massive amount of data to find the true positive matches. The method, rudimentary as it is, functions as a binary probe of a lemma’s existence in one window compared to another, sequential or otherwise. This process, particularly when parsing windows of more than a dozen tokens, produces a large number of false positive as a result. Said effect was a major influence in keeping the token window as small as possible while preserving the integrity of the transposition to the greatest extent allowed. The algorithm can both confirm what was learned from close reading and potentially find its own new transpositions. Finding a particular window of tokens based on its place is simple, though the remaining transpositions potentially to be found must be evaluated by a real person for their meaningfulness. Knowledge of the contexts surrounding particular words is thus still a necessary human-in-the-loop intervention. A fuller description and specific data are available in the Results section below.

These results and an interactive textual exploration are at the center of the project’s “front-end” or user-facing interface, called here “Augmented Textuality” (AT). The web-based application offers users an interactive portal to the corpus via networked instances of biblical source of vocabulary in Gor’kii’s works and Russian literature in general. These points of religious transposition have been determined through both close reading and the semi-automatic process described above. Each of these connections is home to a unique set of data, which adds to our understanding of both the Orthodox Bible and literature. Using an array of digital tools, we can see the texts jump off the page, so to speak, with multiple layers of historical, artistic, and religious context. When viewing the transpositions through the Bible, we see how each referenced segment, whether verse, name, or narrative, has been borrowed to various ends in Russian literature. Viewing them through the literary works, on the other hand, brings into relief the author’s manipulation of the source and thus shines a light on the transposition’s artistry and message. The data set’s value quickly grows as I add additional the works of other Russian authors, which will help us identify hot zones of inspiration from the Bible across literary history.

The AT interface stands atop a simple technical construction of basic internet building blocks. At its foundation is a custom Linux-Apache-mySQL-PHP (LAMP) stack of technologies, while the visual presentation is produced with straightforward HTML/CSS/JS.[[6]](#footnote-7) The decision to create everything from the ground-up, while more time intensive, gives the project a better chance at remaining functional and thus useful as technology changes with time. Linux is an operating system that provides the server environment. Apache is a web server software, which works with other software to manage inputs and “serve” them to the user. The mySQL database is a repository for the transposition objects and their associated data. This input complements the texts themselves by providing analytical information about each instance. Finally, PHP is a programming language that excels at managing pages’ contents on the server, putting them together in the correct format to respond to users’ requests. In this way, a LAMP stack functions like a restaurant with its physical space (Linux) and a waitstaff (Apache), who run errands and information between the storage shelves (mySQL), the chefs (PHP), and the diners (users). The texts, including the Bible and Gorky’s works, are in XML format.[[7]](#footnote-8) This vehicle of information is, to extend the metaphor, the custom plate designed to make the food (the actual analytical content) look and taste its best. Using this file type, I am able to deliver an annotated version with instructions for when and where the computer is to overlay relevant information provided by the database. Success would therefore be like a classic comfort food made even better by an ingredient that we did not know we wanted. Functions like the informational window with transposition analysis were created in standard JavaScript without third-party frameworks.

What I have called the “transposition object” is a central concept that emerges out of the creation of Augmented Textuality. It forms at the intersection of the origin text and its derived transpositions. Like a gem with many facets, the object is composed of multiple interpretations that begin to play off of each other. For example, the sex worker in *Confession* as a transposition shares an edge with the original referent(s) in the Bible while simultaneously sharing edges with preceding transpositions, namely Doestoevskii’s and Tolstoi’s penitent prostitutes. As such, we can characterize each transposition object by its name, type, textual inheritance, and historical-semiotic components. These components are “element,” “continuity,” “function,” “religion,” and “politics.” “Element” refers to the nature of the borrowed Orthodox culture, such as a character, a narrative, or a symbol. “Continuity” refers to the extent the element underwent change at the hands of the transposing author. While each transposition object is likely to have its own unique combination of these qualities, as a collective they can be organized into a Venn diagram or force-directed graph to visualize their similarity to and difference from one another.[[8]](#footnote-9) A complete vocabulary with definitions is available in the project schema (Appendix A). Collecting and recording a significant quantity of transposition objects will open avenues to asking questions about the Russian literary treatment of the Bible across centuries.

Results

Before discussing the more granular data, it may be useful to take a glimpse at the bigger picture of the overlap between Gor’kii’s works and the Orthodox Bible. There are exactly 675,068 tokens in the Russian Synodal Bible, the forms of 20,933 distinct lemmas (~3.1% of tokens are unique). In *Mother*, we find 86,009 tokens and 8,197 lemmas (~9.5%). *Confession* contains 47,484 tokens consisting of 5,989 unique lemmas (~12.6%). The increased lexical diversity reflect of both a varied subject matter and a shorter, therefore less repetitive overall length. Among those sublists, ~49% is shared between Mother and the Bible, while Confession and the Bible share ~54% of a common language. In addition to being shorter, Confession directly addresses religious matters, so it produces an expected result.

In addition, I juxtaposed the top twenty-five lemmas by quantity of the Bible and each novel. *Figure A* shows the most common lemmas for *Mother*, the Bible, and *Confession*, respectively. We can see evidence of at least a broad relationship between the texts where there is a high level of correlation between meaningful co-occurrence of lemmas. As an example, “mother” [*mat’*], as the novel’s top word is relatively unpopular (137th) in the Bible, but its twenty-second most common word, “son” [*syn*] is near the top (third) of the Bible’s list. *Mother* is also notable for its heavy relative emphasis on “hand” [*ruka*], the fifth most common in the novel but only twelfth in the Bible, a nod to the significance of manual labor for Gor’kii.Similarly, “person” [*chelovek*], “God” [*bog*], “land” [*zemlia*], and “being” [*byt’*] are numbers second, fourth, sixth, and thirteenth in *Confession*’s top lemmas, which are tenth, fourth, sixth, and sixteenth, respectively, by frequency in the Bible. By this metric, Confession significantly overlaps with the Orthodox Bible in its entirety, distinguished by its heightened focus on the human with some added interest in existence—or existentialism. A little down the rankings at number eighteen in Confession is “*narod*,” relatively much higher in the Bible at seventh. The *narod* and its religious importance in this context look like an unspoken connotation readers are expected to understand.

Compared to the Bible, both novels overrepresent and underrepresent a number of prominent symbols. Foremost is “person,” which is tenth in the Bible, third in Mother, and second in Confession. The works also simultaneously feature “speak” [*govorit’*] (fourth in *Mother*, first in *Confession*, and fifth in the Bible) and “word” [*slovo*] (sixteenth in *Mother*, fourteenth in *Confession*, and eighteenth in the Bible) more prominently, as well. A generous interpretation of this may be that the data indicate some semblance of a democratization being projected in Gor’kii’s works. At the same time, both Gor’kii’s works underrepresent “tsar,” the eighth most common lemma in the holy book. Individually, *Mother* represses “lord” [*gospod’*] (?, as opposed to first in the Bible), “God” (?, as opposed to fourth), and “father” [*otets*] (?, as opposed to fourteenth). *Confession* deemphasizes [*syn*] (?, as opposed to third), “home” [*dom*] (?, as opposed to eleventh), and “city” [*gorod*] (?, as opposed to seventeenth). In this choice, we see Gor’kii’s earlier novel attempting to avoid fathers of all kinds, which I explained more thoroughly in Chapter 3. Similarly, the latter work’s more mature tone and vagabond theme shine through here. *Figure B* shows these same trends in the negative by putting the light on the words of the Bible the appear most commonly in *Mother* and *Confession*.

In the token window experiment, there was success in identifying known transpositions, but I was unable to discover a new instance with the help of the program. In the end, it was most effective to use a token window of seven to eight tokens. With higher quantities, false positive begin multiplying seemingly exponentially, and below that point, numbers begin dropping rapidly. With seven-token windows, the program identified 45,597 windows with at least 3 overlapping words, with the highest being six of the seven tokens matching. These were, however, examples of the noisiness often encountered in exploratory textual analytics. *Figure C* shows a parsing table I created in order to work through the data. Eliminating stopwords and reducing remaining tokens to lemmas renders windows that can be repetitive, senseless, and vague. The example in the figure demonstrates how simple it may be to actually have six word matches without actually meaning anything significant, like window #5232 from *Mother*. For future purposes in refining the program, more advanced techniques that include semantic context will be crucial to honing the algorithm’s work. Relying on authors to use the exact language, even when adding some flexibility with by deriving tokens’ lemmas, creates more problems than it solves. Additional approaches can be implemented to consider matches between windows based on similarity in meaning rather than the word itself.

Schema

* 1. transposition object
     1. **id** :: brief unique name
     2. **type** :: a typology of the object
        + - pastiche : a friendly, generally faithful recreation an Orthodox referent
          - imitation : a neutral re-creation of an Orthodox referent
          - parody : a mocking, largely negative re-creation of an Orthodox referent
     3. **description** :: author’s use of the Bible in their own work for secular purposes
     4. components
        + **element** :: literary constitution of the transposition
          - character : an entity, usually a person, from the Christian tradition, e.g., Job
          - narrative : a recognizable narrative from an Orthodox text, e.g., parables, hagiographies
          - symbol : a figure, shape, sign, color, or other phenomenon that derives its social meaning from the Christian tradition
          - theme : an identifiable, unifying topic that plays a notable role in Christianity
        + **durability** :: how the original message has been preserved/changed
          - unchanged :
          - changed\_but\_clear : identity of the source
          - changed\_categorically :
          - abstract\_reference :
          - syncretic\_blend : repackage of religious element with significant presence of another ideology, e.g., paganism, tolstovstvo, materialism
        + **religion** :: religious message in the transposition
          - pro-orthodoxy : promoting the
          - anti-orthodoxy : arguing against Orthodoxy as-is, advocating reform but not necessarily abolition
          - anti-established\_religion : arguing against established religion regardless of denomination
          - pro-personal\_religion : arguing in favor of a private, internally held belief system
        + **politics** :: political message in the transposition
          - monarchist : supportive of the Russian Imperial monarch as a theocratic ruler
          - anarchist : supportive of
          - revolutionary :
          - westernizer :
        + **function** :: pragmatic purpose of the transposition in the text
          - social\_critique :
          - political\_critique :
          - church\_critique :
          - moral\_instruction :
          - humor :
     5. **related\_to**
        + agent : may be related to 1+ agents
        + work : may be related to 1+ works
        + bible verse: may be related to 1+ bible verses
  2. **agent**
     1. id ::
     2. type ::
     3. name ::
     4. alternative\_names ::
     5. related\_to
        + work : may be related to 1+ works
        + bible verse range : may be related to 1+ BV ranges
  3. **work**
     1. **id** :: brief unique name
     2. **type** :: a typology of the object
     3. **description** :: author’s use of the Bible in their own work for secular purposes
     4. **pub\_date** :: date of publication
     5. **related\_to**: agent, work, bible\_verse\_range
  4. **bible verse range**
     1. id :: brief unique name
     2. start :: the first chapter and verse in the range
     3. end :: the last chapter and verse in the range
     4. related\_to:
        + work : may be related to 1+ works

1. The underlying programming code for this project as described in this chapter is available at <https://github.com/kollektivminds/russian-literary-bible>. [↑](#footnote-ref-2)
2. “В глубокой древности, когда человек начал мыслить, он мыслил технологически, то есть опираясь исключительно и только на свой трудовой опыт. Технология — это логика фактов, создаваемых трудовой деятельностью людей, идеология — логика идей, то есть логика смыслов, извлечённых из фактов, - смыслов, которые предуказуют пути, приёмы и формы творчества новых фактов.” [↑](#footnote-ref-3)
3. “Publications,” Yandex Research, accessed March 1, 2025, <https://research.yandex.com/publications>. [↑](#footnote-ref-4)
4. DP: https://github.com/deeppavlov

   Natasha: https://github.com/natasha [↑](#footnote-ref-5)
5. To compare the outcomes, we may look [↑](#footnote-ref-6)
6. Technical jargon is vast and ever-changing. The particulars of each technology are not important for the present purposes. Nevertheless, a gloss: Standard Query Language (SQL), a database-oriented programming language; PHP Hypertext Preprocessor (PHP), a server scripting language for web development and recursive acronym; HyperText Markup Language (HTML), a language to structure webpages and their content; Cascading Style Sheets (CSS), a styling language to programmatically tag and design web elements; JavaScript (JS), a scripting language to dynamically manage data and page content. [↑](#footnote-ref-7)
7. eXtensible Markup Language (XML), a language in the same family as HTML, used to structure and integrate different types of data in a custom, transmittable document environment. [↑](#footnote-ref-8)
8. A force-directed graph is a computational method of simulating forces of attraction and repulsion to nodes (items) in a networked relationship. Elements are first spread out evenly, and then are “pulled” and “pushed” by other nodes based on a pre-defined set of criteria. This algorithmic abstraction allows researchers to examine a network’s reactions to a variety of variables in a reproducible way. [↑](#footnote-ref-9)