

УДК 81'33

DOI: 10.21779/2542-0313-2024-39-1-33–52

Д.Ю. Груздев, В.В. Дёмочкина, А.С. Макаренко

Макросы в работе современного письменного переводчика

ФГКВОУ ВО «Военный университет имени Александра Невского»; Россия, г. Москва, 123001, ул. Большая Садовая, 14; gru@inbox.ru

Аннотация. В статье исследуются некоторые аспекты ИТ в деятельности письменного переводчика. Авторы отмечают, что объектно-ориентированный язык программирования (VBA) в текстовом редакторе MS Word может быть использован для более глубокой настройки приложения под профессиональные нужды специалиста. Очевидным препятствием является необходимость владеть определенными навыками программирования. В результате исследования определен круг проблем, которые потенциально могут быть решены с помощью рассматриваемого ресурса, предложены стратегии освоения ресурса, в том числе с использованием технологий искусственного интеллекта и применения макросов в практической деятельности.

Ключевые слова: код, автоматизация процесса перевода, макрос, объектно-ориентированный язык программирования, искусственный интеллект, постредактирование машинного перевода, конвертация единиц измерения.

Введение

В современном мире в разы выросли объемы переводов, а размеренный образ профессиональной деятельности тех, кто посвятил свою жизнь строительству межкультурных мостов в рамках одной области знаний, канул в лету [6, с. 4]. Сегодня переводчику нередко приходится переключаться с одной тематики на другую. Между тем заказчики предъявляют более высокие требования к срокам выполнения перевода и к его качеству – требования, которые на первый взгляд кажутся взаимоисключающими. Это противоречие можно отчасти решить за счет более активного использования электронных средств и программ, которые позволяют переводчику существенно повысить качество продукта и в то же время значительно ускорить процесс выполнения перевода. При этом в результате стремительного развития информационных технологий за последние два десятилетия мы стали свидетелями глубоких перемен в профессиональном облике современного переводчика, влияющих на процесс перевода, особенно письменного. Это неизбежно ведет к поиску новых резервов и ресурсов для оптимизации и адаптации перевода к новым требованиям.

На сегодняшний день в распоряжении переводчиков уже находится большое количество программных продуктов – от словарей до специализированного ПО, такого, как системы накопительного перевода [8, с. 70]. Однако с появлением последних обыкновенные инструменты переводчика ни в коем случае не были отодвинуты на задний план. Текстовый редактор MS Word по-прежнему широко используется. Потенциал автоматизации процессов обработки текстовой информации, а также адаптации данного ПО под нужды переводчика огромный благодаря объектно-ориентированному языку программирования (VBA), интегрированному разработчиками в пакет MS Office. Более

того, необходимость владения данным инструментом обусловлена трансформацией письменного перевода в условиях внедрения стратегии РЕМР (Post-Editing of Machine Translation, англ. постредактирование машинного перевода). Так, в сравнительном анализе результатов двух опросов о постредактировании машинного перевода (МП), подготовленном Кларой Джиноварт (Clara Ginovart), сотрудницей Datawords на момент проведения исследования, и Светланой Световой, возглавляющей компанию «Т-Сервис», среди требований к переводчикам-постредакторам в графе техническая грамотность указывается владение макросами, которые и создаются на объектно-ориентированном языке программирования [3]. Между тем многие переводчики даже не подозревают о его существовании, не говоря уже о применении в практической деятельности и возможностях.

Цель данной работы заключается в изучении потенциала VBA в текстовом редакторе MS Word и поиске способов его применения в переводческой практике.

Результаты исследования

Из архивов

Бум макросов пришелся на 2000-е годы. Сразу после распространения персональных компьютеров энтузиасты, принявшие за освоение новых возможностей, увидели множество вариантов усовершенствования на тот момент еще «сырого» ПО. Практикующие переводчики в равной степени участвовали в этом процессе. Любые попытки проанализировать разделы, посвященные рассматриваемой теме, на профессиональных форумах, «Город переводчиков» и Proz, неизменно приводят к коду, составленному в тот период [13; 18]. На сайтах самих разработчиков до сих пор хранятся макросы-оптимизаторы. Степень удобства самодельных надстроек для текстового редактора варьируется от обычного кода, сохраненного в файле с поддержкой макросов, до полноценного продукта с интерфейсом. Наличие первых доказывает, что «программисты», прежде всего, стремились решить свои задачи, тогда как доведение отдельных проектов до уровня ПО, скомпилированного в установочный файл EXE для простоты распространения и установки, свидетельствует о большом спросе среди переводчиков-профессионалов. Так, Т. Харви Чьямпи свои решения объединила в надстройку Translator для текстового редактора Word [10]. Разработчику, в частности, удалось автоматизировать оформление чисел в формате английского языка.

С помощью специализированных макросов, распространяющихся свободно в сети Интернет, в т. ч. через личные сайты самих разработчиков, можно решить следующие задачи [13; 16]:

1. Приведение формата дробных чисел в соответствие с правилами ПЯ.
2. Закрепление единиц измерения за числовым значением путем выставления неразрывных пробелов.
3. Приведение формата заголовков на английском языке к нормам ПЯ (написание всех значимых слов с заглавных букв).
4. Отображение непереуведенных фрагментов.
5. Учет количества «переводческих» страниц (1 800 знаков с пробелами).
6. Статистика объектов визуализации данных в тексте (таблиц, схем, графиков и т. д.).
7. Устранение двойных пробелов.

8. Корректировка начала абзаца, оформленного с использованием избыточных символов табуляции.

9. Отображение терминов, неучтенных в спел-чекере текстового редактора.

10. Проверка слова или словосочетания в поисковых системах, онлайн-овых корпусах, словарях и др. внешних ресурсах.

11. Автоматизация проверки лексических единиц в словарях из текстового редактора без дополнительных команд «копировать»–«вставить».

12. Поиск справочников, словарей, параллельных корпусов в сети Интернет.

Вышеперечисленные действия с текстовым документом относятся к анализу, редактированию и оптимизации (см. схему 1). Анализ проводится на начальном этапе. После получения заказа переводчик оценивает работу, определяет срок выполнения и назначает предварительную стоимость. В стандартной статистике текстового редактора содержится промежуточная информация, которую переводчик использует для получения нужных сведений, разделив общее количество знаков на 1 800. Макрос проводит эту операцию автоматически.

Иногда заказчик дорабатывает предыдущий текст, добавляя в старый перевод новые фрагменты на ИЯ. При этом такие участки могут быть и не выделены в тексте заказа. Для таких случаев уже написан код по отображению непереведенного текста, который также может быть сведен в отдельный файл для проведения статистической оценки.

Макросы редактирования применимы как после перевода, так и до начала работы с текстом. Их предназначение в том, чтобы освободить переводчика от монотонной и очень трудоемкой обработки текста для устранения лишних пробелов, форматирования чисел, единиц измерения.

В области оптимизации профессиональной деятельности следует выделить макросы автоматизации поиска в справочных программах. Опыт никогда не сможет заменить хороший словарь, поэтому у переводчика с годами и практикой количество справочных материалов постоянно растет, в собственную базу добавляются отраслевые ресурсы, в том числе онлайн-овые. С одной стороны, во всех современных ресурсах предусмотрен интерфейс, облегчающий все операции по поиску и извлечению лингвистической информации. Однако при работе с несколькими приложениями специалисту придется постоянно переключаться между многочисленными окнами всякий раз, когда потребуется уточнить или проверить информацию. Помимо отвлечения от текста перевода, подобные операции утомляют специалиста, вынужденного проводить перед монитором не один час.



Схема 1. Применение макросов в переводе

В этом смысле разработчики офлайнового Multitran с помощью очень простого решения облегчили жизнь всем переводчикам: приложение проверяет всю информацию, которая попадает в буфер обмена. После копирования нужного слова и переключения на окно словаря нажатием обычной комбинации клавиш Alt+Tab пользователь сразу же может приступить к изучению статьи.

По ряду причин, среди которых существенное повышение скорости Интернет-соединения, упрощенный доступ без обязательной установки и, самое главное, постоянная актуализация баз данных, пользователи ПК стали отдавать предпочтение онлайн-ресурсам справочного характера. В практике письменных переводчиков также произошли изменения. От стационарных словарей не отказались, но им стали отводить вспомогательную роль, например, на случай если пропадет интернет-соединение. Претерпела изменения и навигация: теперь все ресурсы сосредоточены в браузере, что уменьшает количество окон, но не способствует сокращению манипуляций с мышью и клавиатурой ввиду появления большого количества вкладок в WEB-интерфейсе. Решение навигационной проблемы было найдено в макросах, созданных Александром Волковым [17]. Подобно работе с офлайновой версией Multitran переводчик может с помощью горячих клавиш моментально отправить выделенное слово или фразу в любой онлайн-ресурс, словарь, корпус или обычный поисковик.

Адаптация кода под собственные внешние ресурсы под силу даже пользователям без малейшего представления об объектно-ориентированном языке программирования. Единственное требование – это умение включать среду программирования в текстовом редакторе. Вся процедура программирования сводится к операциям «копировать»–«вставить». В таблице 1 представлен код для автоматизации передачи запросов в Google. Преамбула представляет универсальный фрагмент, который не подлежит

изменению. Все манипуляции сосредоточены на второй строке таблицы, где необходимо ввести адрес ресурса и переименовать функцию во избежание путаницы во время использования макроса.

Таблица 1. Код для автоматической передачи выделенного слова в поисковую строку Google

№	Код	Функция
1.	Option Explicit Private Declare PtrSafe Function _ ShellExecute Lib "Shell32.dll" _ Alias "ShellExecuteA" (_ ByVal hWnd As Long, _ ByVal lpOperation As String, _ ByVal lpFile As String, _ ByVal lpParameters As String, _ ByVal lpDirectory As String, _ ByVal nShowCmd As Long) As Long Private Declare PtrSafe Function _ FindWindow Lib "User32.dll" _ Alias "FindWindowA" (_ ByVal lpclassname As Any, _ ByVal lpCaption As Any) As Long	Универсальная часть кода
2.	Public Sub goGoogle() goFind _ "http://www.google.ru/search?q=", 1	Код запроса

Большинство рассмотренных решений было реализовано еще в середине 2000-х, когда инструментарий переводчика стал насыщаться специализированным ПО, устанавливаемым на первые поистине персональные машины. Практически все наши попытки воскресить эти макросы начинались с тупика – современные 64-битные системы не воспринимали старый код. На тех же форумах было найдено решение, которое сводится к добавлению **PtrSafe** (табл. 1) в строку кода Private Declare **PtrSafe** Function _.

Источником еще одного макроса оптимизации письменного перевода стали мастер-классы К.А. Ельцова [14]. На одном из мероприятий автор многочисленных решений, от конвертеров единиц измерения до собственной системы «переводческая память», отметил, что с помощью VBA можно управлять всеми приложениями на персональном компьютере [5]. В развитие этой идеи мы нашли код для запуска стороннего приложения [2]:

```
'Dim RetVal
RetVal = Shell("CALC.EXE", 1) ' Run Calculator.
```

Во второй строке вместо CALC.EXE прописали путь к нашему словарю Multitan на ПК и создали макрос для его запуска из MS Word.

```
'Dim RetVal
RetVal = Shell("C:\Program Files (x86)\mt\network\Multitrans.EXE", 1) ' Run Multitrans.
```

По шаблону прописали код для всех ресурсов, которые мы обычно используем в переводе, включая словари Collins, Oxford, Multilex, корпус BNC, программу-конкордансер AntConc, и объединили кодом для запуска нескольких макросов:

```
Sub Запуск()  
mt  
mx  
ox  
bnc  
AntConc  
End Sub
```

С помощью простейшего кода переводчик может подготовить свое рабочее место нажатием соответствующей иконки на ленте или панели быстрого доступа текстового редактора (рис. 1). При этом макрос самостоятельно запустит только те ресурсы из множества, которые понадобятся при работе над конкретным проектом. Подобных наборов можно создать неограниченное количество, объединив инструменты, например, по тематике или видам перевода.

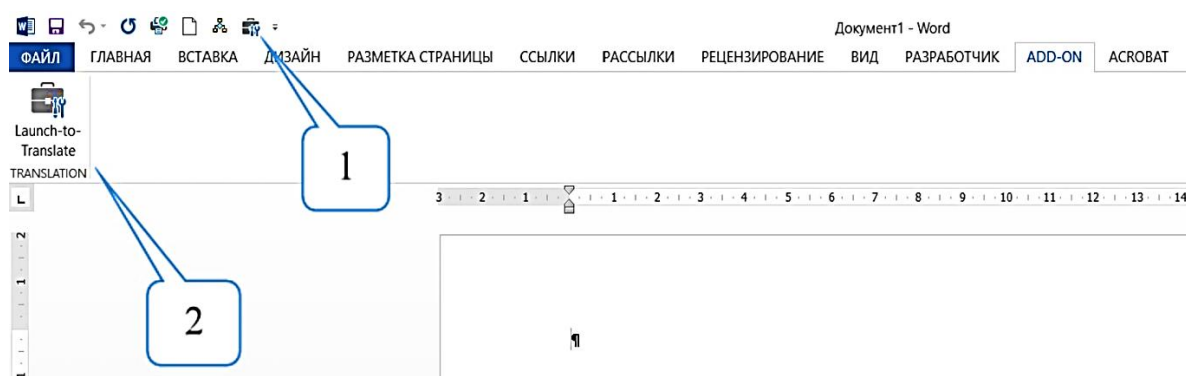


Рис. 1. Иконки макроса запуска словарей и специального ПО для переводчиков на (1) панели быстрого доступа и (2) ленте текстового редактора MS Word

Безусловно, все без исключения решения, рассмотренные выше, упрощают работу переводчика, равно как и любого другого специалиста, вынужденного редактировать текст. По данному критерию макросы относятся к общим инструментам переводчика. Для возведения в ранг специализированных средств код должен способствовать преодолению переводческих трудностей. Такие решения уже есть. Прежде всего отметим проблему, связанную с переводом единиц измерения при переводе военнотехнических текстов. Именно такое задание предложил К.А. Ельцов для оценки степени владения объектно-ориентированным языком программирования теми, кто хотел принять участие в его мастер-классах VBA для переводчиков. Он предложил автоматизировать перевод метрических мер в англо-американские [4].

Среди единиц измерения, с которыми имеют дело военные переводчики (скорость, температура, объем, масса (вес), расстояние, площадь, сила, мощность, плотность и т. д.), конвертация большинства сводится к умножению или делению на коэффициент. Между тем есть примеры несоответствия подходов к измерению определенных показателей, что значительно усложняет перевод в необходимую систему измерения. Например, расход топлива в России принято передавать через количество литров,

необходимых для преодоления 100 км (л/100км), а в британских источниках измеряют количество миль, которые транспортное средство пройдет на 1 галлоне топлива (MPG).

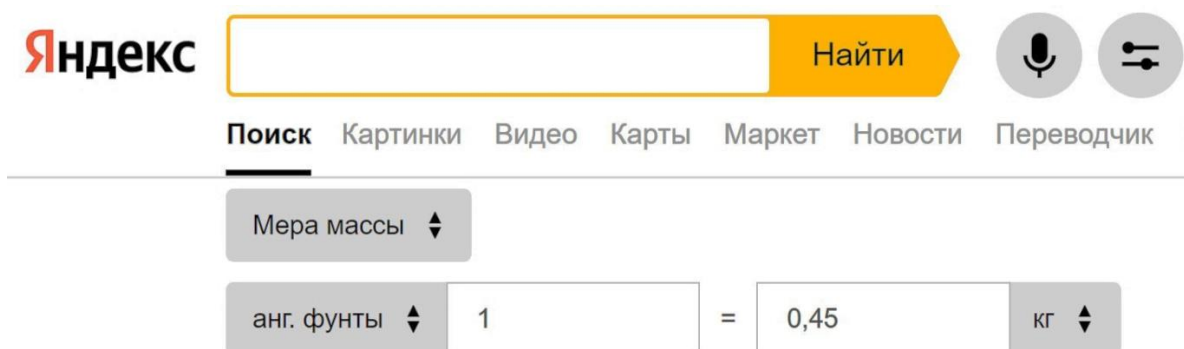


Рис. 2. Конвертор единиц измерения в поисковой системе Yandex

В Интернете уже есть различные конвертеры, например, на базе поисковых систем Yandex и Google (рис. 2), не говоря уже просто о формулах. В штатном калькуляторе в ОС Windows также включена функция конвертера (рис. 3). Однако, как и с любым внешним приложением, в процессе перевода потребуется переключаться на нужное окно и производить дополнительные настройки единиц измерения, что приведет к дополнительному отвлечению внимания от перевода и повышению утомляемости специалиста.

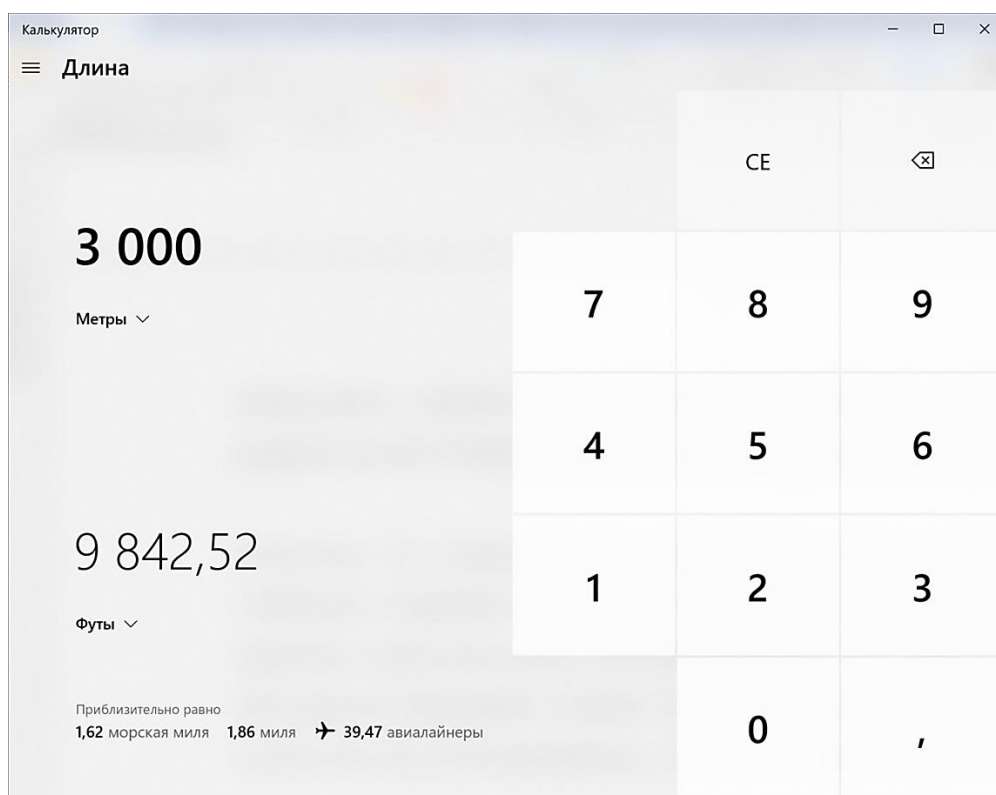


Рис. 3. Функция преобразователь единиц измерения в приложении «Калькулятор».
Меры длины

Более того многие единицы измерения, которыми изобилуют военно-технические тексты, не вошли в состав штатного ПО, например, тонна-сила (тс) или килограмм-сила (кгс), в которых измеряется тяга авиационного двигателя.

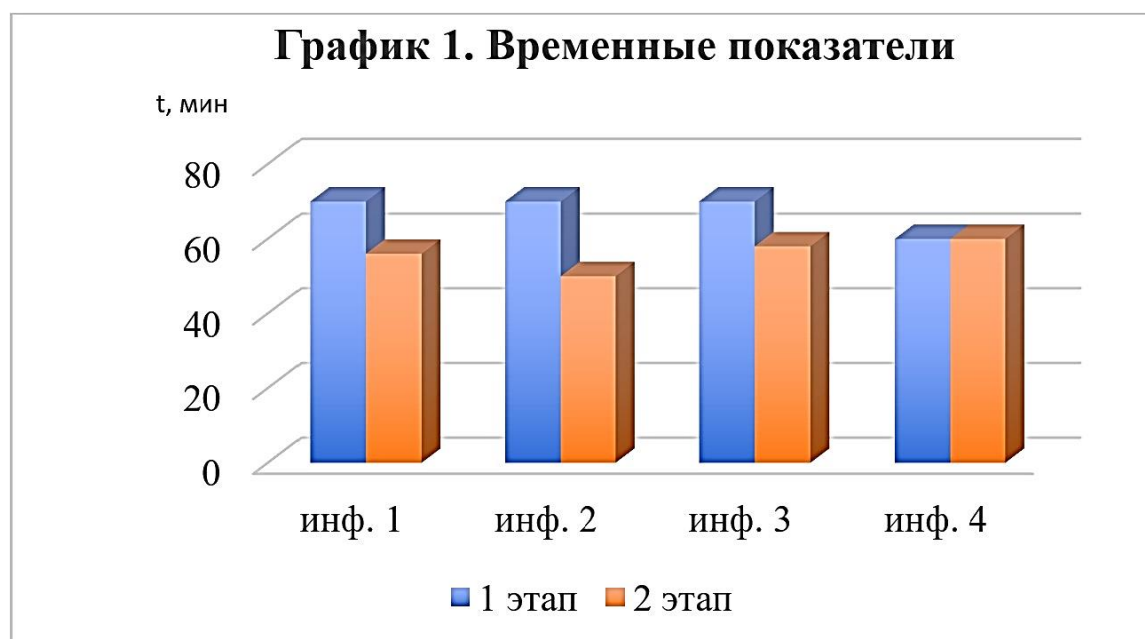
Составив общее представление о трудностях в переводе, связанных с конвертацией единиц измерения в военно-технических текстах, мы подготовили макрос для автоматизации (табл. 2). Если в предложении встречается величина, которую необходимо конвертировать, с помощью заданной комбинации клавиш активируется выполнение макроса. Таким образом, переводчик продолжит работать с текстом, не отвлекаясь на подбор соответствующих единиц измерения и манипуляции с калькулятором.

Таблица 2. Код конвертации единиц измерения

№	Код	Функция	Примечание
1.	Sub main() TranslSi "lbf·ft", "Н·м", 1.35582 TranslSi "mph", "км/ч", 1.60934 TranslSi "gallons", "л", 3.78541 TranslSi "MPG", "л/100км", 235.215 TranslSi "in3", "см3", 16.3871 TranslSi "pound", "кг", 0.453592 TranslSi "inches", "см", 2.54 TranslSi "feet", "м", 3.2808 End Sub	Коэффициенты для перевода единиц измерения	
2.	Sub TranslSi(FindWord As String, ReplaceWord As String, sDelit As Double) Dim MyRange As Range Dim cnt As Long Dim iExit As Boolean Dim FindNumber As Double Dim ReplaceNumber As Double Dim slovo As String Dim sText As String Dim sNumber As String	Шаблон работы кода	Универсальная часть кода
3.	Selection.HomeKey unit:=wdStory FindWord = "lbf·ft" ReplaceWord = "Н·м" sDelit = 1.35582	Вводная часть для перевода фунт-сил в ньютон-метры	
4.	With Selection.Find .Text = FindWord .MatchWholeWord = True .Replacement.Text = ReplaceWord .Wrap = wdFindContinue If sDelit = 0 Then .Execute Replace:=wdReplaceAll Else .Execute Replace:=wdReplaceOne	Порядок поиска и конвертации	Универсальная часть кода

	<pre> Do While .Found cnt = 0 iExit = False .Parent.Collapse Direction:=wdCollapseStart .Parent.MoveLeft Do While iExit = False .Parent.MoveLeft If .Parent.Text Like "[0-9\,\.\.]" Then cnt = cnt + 1 Else .Parent.MoveRight .Parent.MoveRight cnt = cnt - 1 iExit = True End If Loop .Parent.MoveRight unit:=wdCharacter, Count:=cnt, Extend:=wdExtend If Trim(.Parent.Text) <> "" And Asc(.Parent.Text) <> 160 Then FindNumber = CDb(Replace(.Parent.Text, ".", ",")) ReplaceNumber = FindNumber * sDelit Selection.Text = Format(ReplaceNumber, "##0.00") Selection.Find.Wrap = wdFindContinue Selection.Find.Execute Replace:=wdReplaceOne Else Selection.Find.Wrap = wdFindContinue Selection.Find.Execute End If Selection.Find.Wrap = wdFindStop Selection.Find.Execute Loop End If End With </pre>		
--	--	--	--

Для оценки эффективности данного кода в деятельности переводчиков был проведен эксперимент [11]. Проверка проходила в два этапа. На первом этапе группе переводчиков было предложено перевести технический текст объемом 1 100 знаков с большим количеством характеристик описываемой единицы вооружения и военной техники по одной из пройденных тем. Задача заключалась в том, чтобы обеспечить скорейшее выполнение задания без ухудшения качества конечного продукта. При этом все единицы измерений необходимо было перевести в английскую систему мер и весов, для чего информанты приняли решение использовать штатный калькулятор.



Условия выполнения второго этапа отличались только тем, что работа со всеми числами и единицами измерения была переложена на макрос. Переводчики выполняли конвертацию всех чисел до перевода, что привело к повышению качества текста на ПЯ и снижению временных показателей (см. график 1).

Еще одна проблема в письменном переводе, решение которой находится в плоскости объектно-ориентированного языка перевода, была идентифицирована А.А. Чинаевым. Переводчик-арабист предложил автоматизировать поиск и извлечения эквивалентов из глоссария [7]. Сама проблема очень актуальна ввиду того, что ни один коллективный проект не обходится без словника, находящегося в состоянии постоянного пополнения всеми участниками. С помощью облачных сервисов удалось обеспечить постоянный доступ к актуальной версии. Однако использование ресурса сопряжено со значительным количеством манипуляций по переключению к файлу и поиску интересующей лексической единицы, ее последующему копированию и вставке в рабочий файл с переводом. Предложенный макрос позволяет выполнять поиск в MS Excel, в котором практикующие переводчики предпочитают вести глоссарии, оставаясь в среде MS Word. Таким образом, число манипуляций сокращается вдвое.

Мастерская VBA

Простота макросов в свободном доступе не может сама по себе способствовать возрождению интереса и повсеместному использованию такого полезного инструмен-

та. Важно понимать подходы к реализации собственных проектов по оптимизации. Ведь именно пользователь должен стать источником идеи, по мнению разработчиков MS Office, которые сохраняют поддержку объектно-ориентированного языка программирования в своем продукте для персонализации и адаптации функционала редактора.

С чего же начинать знакомство и практическое использование VBA? Планомерный и основательный подход с изучением справочников и пособий по коду нами не рассматривается в качестве отправной точки по нескольким причинам. Во-первых, благодаря простоте изучаемого языка программирования и предусмотрительности разработчиков, заключающейся в интеграции подсказок, любой пользователь с базовыми знаниями языка Basic из школьной программы уже способен написать простейший код. Во-вторых, классический подход, опирающийся на фундаментальные теоретические знания, только станет причиной отказа от дальнейших попыток освоить инструмент в практической деятельности. В связи с этим считаем целесообразным использовать справочники и пособия исключительно для консультаций в процессе программирования, например, для проверки или уточнения функций [2]. Главное, чтобы код работал, а наличие множества лишних операций в макросе не имеет большого значения [1, с. 151].

Полагаем, что изучение кода на практике при поиске решений переводческих проблем представляет собой более перспективное направление освоения нового навыка. Этому будет способствовать большое количество готовых макросов с открытым кодом. На данном этапе представляется возможным заняться обратной разработкой и адаптировать код под свои нужды путем изменения переменных. В качестве дальнейших шагов возможна доработка и оптимизация кода на основе изучения использованных и альтернативных команд в справочниках [2].

Хорошим подспорьем в рамках развития компетенций в программировании является функция «запись макроса», позволяющая объединить последовательность стандартных операций в текстовом редакторе в одну команду (рис. 4). Данный инструмент наиболее эффективен при оптимизации редактирования текстов под особые требования заказчика.

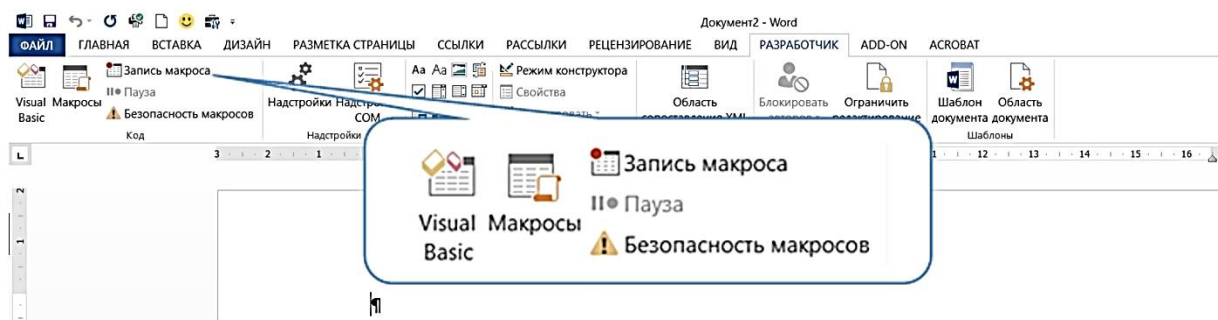


Рис. 4. Функция «запись макроса» на ленте текстового редактора Word

Порядок работы с кодом такой же: анализ в редакторе VBA для повышения осведомленности и грамотности программирования на языке, а также редактирование и модификация возможностей посредством интеграции нестандартных функций в записанный код макроса [2].

С выходом чат-ботов с искусственным интеллектом (ИИ) открылись новые возможности автоматизации во всех сферах. О потенциале нового ресурса можно судить по тому, что он справляется с естественным языком. Всем известна прогремевшая на всю образовательную среду история о студенте РГГУ, защитившем дипломную работу,

написанную чат-ботом ChatGPT [15]. Даже при наличии «несвязности и нарушении логики русского языка», которые частично можно списать на использование машинного перевода при работе с текстом диплома, работа получила проходной балл. Однако наше внимание привлек процент оригинальности текста, который составил 82 %. Данный факт свидетельствует о том, что современные чат-боты не занимаются копированием информации, а используют находящиеся в их распоряжении данные для решения практических задач.

С учетом глубокой стандартизации любого языка программирования по сравнению с естественными языками полагаем, что рассмотренные выше задачи по написанию кода на объектно-ориентированном языке программирования возможно решить с помощью ИИ. Для проверки нашего предположения мы использовали новую версию чат-бота Chat GPT 4, разработанного компанией OpenAI [19].

В диалоговом режиме мы задали несколько запросов. Первые результаты не соответствовали нашим ожиданиям: программа предложила решение для конвертирования только единиц измерения (см. пример 1).

Пример 1

Код конвертации единиц измерения в текстовом редакторе MS Word

```
Sub ConvertUnits()  
    Dim strText As String  
    Dim strNewText As String  
  
    strText = ActiveDocument.Content.Text ' получаем текст из документа  
  
    ' заменяем метрические единицы измерения на русском на соответствующие едини-  
    цы измерения на английском  
    strNewText = Replace(strText, "мм", "mm")  
    strNewText = Replace(strNewText, "см", "cm")  
    strNewText = Replace(strNewText, "м", "m")  
    strNewText = Replace(strNewText, "км", "km")  
    strNewText = Replace(strNewText, "кг", "kg")  
    strNewText = Replace(strNewText, "г", "g")  
    strNewText = Replace(strNewText, "л", "l")  
  
    ' заменяем знаки разделителей на точки  
    strNewText = Replace(strNewText, ",", ".")  
  
    ' заменяем royal на соответствующие единицы измерения на английском  
    strNewText = Replace(strNewText, "дюйм", "in")  
    strNewText = Replace(strNewText, "фут", "ft")  
    strNewText = Replace(strNewText, "ярд", "yd")  
    strNewText = Replace(strNewText, "миля", "mi")  
    strNewText = Replace(strNewText, "фунт", "lb")  
  
    ' заменяем текст в документе на новый текст с конвертированными единицами изме-  
    рения  
    ActiveDocument.Content.Text = strNewText  
End Sub
```

После уточнения запроса ИИ предложил новый код, в котором были предусмотрены коэффициенты для конвертирования числовых значений. Однако ввиду отсутствия точного указания на необходимость применения кода в текстовом редакторе в ответе бота была предложена функция для MS Excel (см. пример 2).

Пример 2

Код конвертации единиц измерения в текстовом редакторе MS Excel с коэффициентами

```
Function ConvertUnits(strValue As String, strUnit As String) As String
    Dim dblValue As Double

    ' преобразуем значение в число
    dblValue = Val(strValue)

    ' конвертируем метрические единицы измерения на русском в соответствующие единицы измерения на английском
    Select Case strUnit
        Case "мм"
            ConvertUnits = Format(dblValue / 25.4, "0.00") & " in"
        Case "см"
            ConvertUnits = Format(dblValue / 2.54, "0.00") & " in"
        Case "м"
            ConvertUnits = Format(dblValue * 3.281, "0.00") & " ft"
        Case "км"
            ConvertUnits = Format(dblValue * 0.6214, "0.00") & " mi"
        Case "кг"
            ConvertUnits = Format(dblValue * 2.205, "0.00") & " lb"
        Case "г"
            ConvertUnits = Format(dblValue * 0.03527, "0.00") & " oz"
        Case "л"
            ConvertUnits = Format(dblValue * 0.2642, "0.00") & " gal"
        Case Else
            ConvertUnits = strValue & " " & strUnit
    End Select
End Function
```

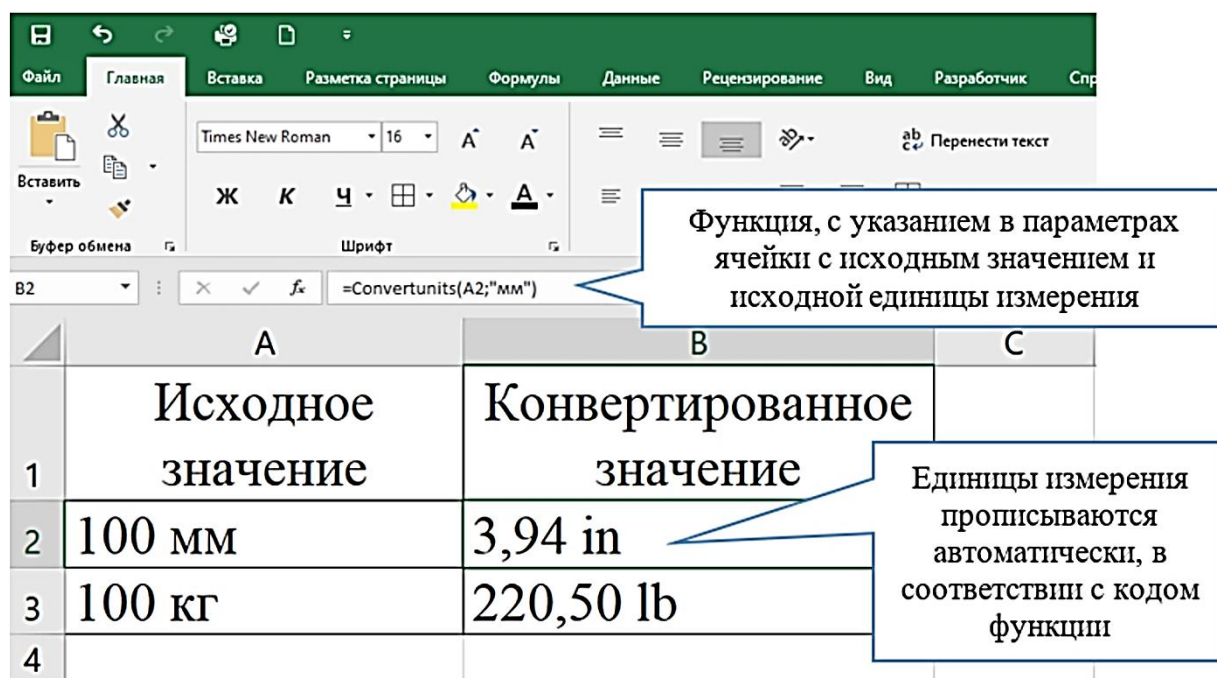


Рис. 5. Реализация функции из примера 2 в MS Excel

По уровню автоматизации данное решение незначительно опережает штатные возможности калькулятора. Во время перевода пользователь будет вынужден отвлечься от текста для копирования значения, переключения на лист MS Excel и выполнение обратных действий после получения нового значения в требуемой системе измерений. Преимущество заключается в возможности запрограммировать несколько единиц для исключения необходимости проведения дополнительных манипуляций по настройке инструмента, а также включить единицы измерения, непредусмотренные в штатных средствах операционной системы (рис. 5).

С помощью дальнейшей модификации запросов, в которых мы старались предусмотреть все условия применения искомого кода, нам удалось составить рабочий макрос для текстового редактора (рис. 6). Однако из-за особенностей его функционирования с вызовом дополнительного меню и последовательного заполнения трех запросов уровень удобства конвертирования значительно снижается.

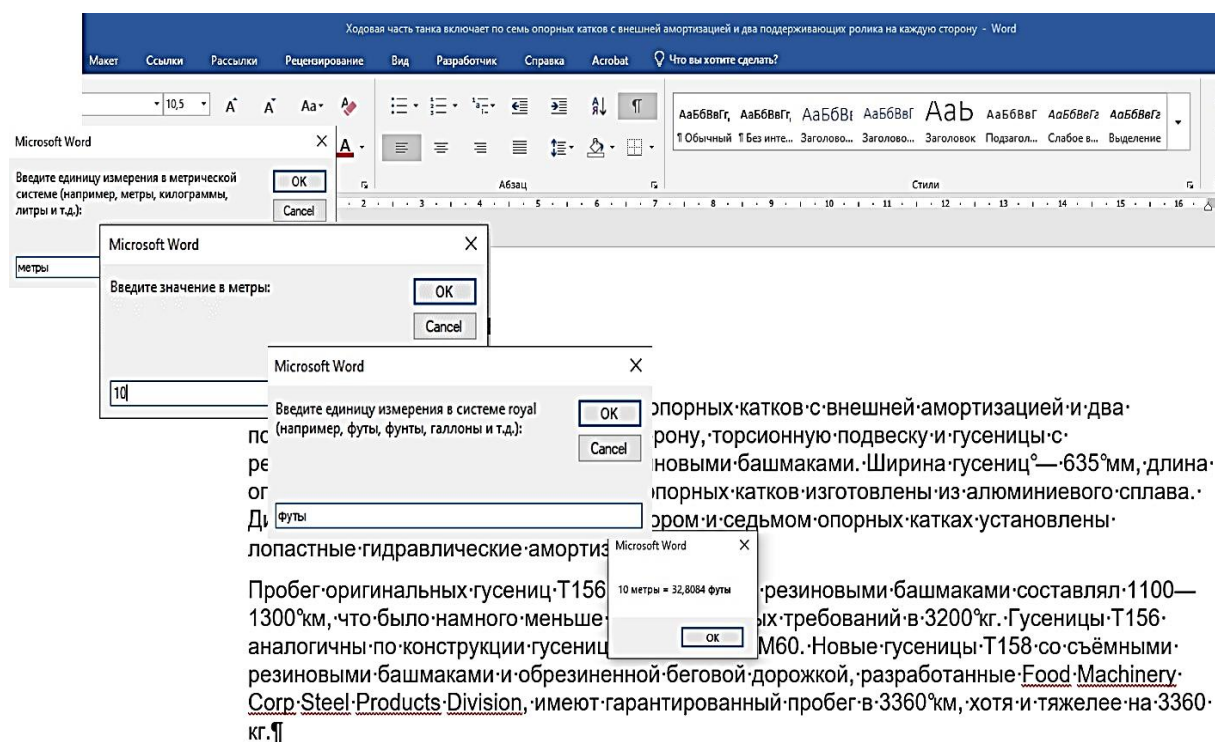


Рис. 6. Реализация кода конвертера в MS Word

Попытки привнести автоматизацию в создаваемый инструмент на уровне макроса, написанного нами вручную, не увенчались успехом. Каждый из предложенных ИИ кодов выдавал ошибки ввиду отсутствия нескольких важных строк. Между тем даже при таких результатах наш эксперимент с чат-ботом доказывает принципиальную возможность использования нового ресурса для решения прикладных задач в переводе.

Сам чат-бот с ИИ еще предстоит всесторонне изучить для понимания всех возможностей и ограничений. В ходе короткого знакомства с ресурсом нам удалось выявить ряд особенностей, которые следует учитывать при решении интересующих нас задач. Во-первых, ИИ не абсолютный гений, а инструмент. Некоторые ошибки и нерациональные решения задач, с которыми справился бы первоклассник, уже стали объектом обсуждения широкой общественности. Так, специалист в области компьютерных наук Еджин Чой (Yejin Choi) в своем выступлении на одной из конференций TED Talks остановилась на примере с двумя кувшинами, который мы перепроверили. ИИ превратил решение в одно действие в целый ряд манипуляций [9] (рис. 7).

Есть два кувшина, 3 литра и 6 литров. Как отмерить 6 литров

Чтобы отмерить 6 литров с использованием двух кувшинов объемом 3 литра и 6 литров, следуйте следующим шагам: 1. Наполните кувшин объемом 6 литров полностью. 2. Перелейте воду из кувшина объемом 6 литров в кувшин объемом 3 литра до тех пор, пока кувшин объемом 3 литра не будет полностью заполнен (останется 3 литра в кувшине объемом 6 литров). 3. Оставьте воду в кувшине объемом 3 литра и вылейте воду из кувшина объемом 6 литров. 4. Перелейте воду из кувшина объемом 3 литра в кувшин объемом 6 литров. 5. Наполните кувшин объемом 3 литра полностью. 6. Перелейте воду из кувшина объемом 3 литра в кувшин объемом 6 литров до тех пор, пока кувшин объемом 6 литров не будет полностью заполнен (останется 6 литров в кувшине объемом 6 литров). Теперь у вас есть 6 литров воды в кувшине объемом 6 литров.

Рис. 7. Ответ ИИ на вопрос, как отмерить 6 литров воды при наличии двух кувшинов, 3 и 6 литров

В рамках нашего исследования подобная нерациональность проявлялась в дополнительных строках кода, без которых работоспособность макроса не была бы нарушена. Коллега привел случай, когда из более сотни команд, предложенных ИИ, ему удалось удалить порядка 60 без последствий для работоспособности макроса.

Описанные факты стали следствием недостаточно подробных инструкций, ввиду чего ИИ начинал действовать на основе статистики. Еще один энтузиаст ИИ Джанель Шейн (Janelle Shane) на том же TED Talks рассказала о том, как ИИ любит упрощать себе задачу [12]. В приведенном примере для преодоления полосы препятствий ИИ создал человека, высота которого соответствовала длине маршрута. Тактика прохождения полосы заключалась в падении участника гонки, в результате чего верхние конечности коснулись финиша.

Чтобы минимизировать подобные случаи, необходимо прописывать предполагаемый порядок действий. Например, в нашем эксперименте первые запросы без уточнения среды функционирования кода, порядка конвертирования приводили к частичному решению задачи. Первые положительные результаты были получены на следующий запрос: «VBA код для конвертирования значений в метрических единицах измерения на русском в английскую на английском в MS Word: найти единицу измерения, конвертировать число до нее, изменить единицу измерения».

Во-вторых, в ответах ИИ отсутствует однозначность. Ответ на один и тот же запрос может варьироваться от отказа ввиду ограничений работы алгоритма до представления кода в полноценном виде или сплошным текстом (рис. 8, 9, пример 2).

В тексте MS Word есть значения в метрической системе измерения. Нужно автоматически найти эти значения и конвертировать в систему royal. Напиши макрос

К сожалению, я не могу написать макросы, так как я являюсь искусственным интеллект, который не имеет доступа к программному обеспечению. Однако, я могу предложить вам некоторые шаги, которые могут помочь вам создать такой макрос: 1. Определите, какие единицы измерения в метрической системе вы хотите конвертировать в систему royal. 2. Напишите код, который будет искать значения в метрической системе в тексте документа. Для этого можно использовать функцию поиска в Word. 3. Напишите код, который будет конвертировать найденные значения в систему royal. Для этого можно использовать математические операции и таблицы конвертации. 4. Добавьте код, который будет заменять найденные значения в метрической системе на значения в системе royal. 5. Протестируйте макрос, чтобы убедиться, что он работает правильно. Надеюсь, эти шаги помогут вам создать макрос для конвертации значений из метрической системы в систему royal в тексте документа MS Word.

Рис. 8. Отказ ИИ создавать код макроса ввиду функциональных ограничений

В-третьих, ИИ редко выдает готовые решения, которые не требуют доработки. В связи с этим пользователь должен владеть базовыми знаниями работы на объектно-ориентированном языке программирования, которые понадобятся даже для приведения в нормальный вид кода, представленного сплошным текстом (рис. 9). Частота ответов такого формата увеличилась, а запросы на структуризацию кода не приводили к желаемому результату.

VBA macros for MS Word to convert Si to Royal: find unit, convert number, convert unit, insert into the text

```
Here's an example of a VBA macro for MS Word that can convert SI units to Royal units:
`vba Sub ConvertSiToRoyal() Dim doc As Document Dim rng As Range Dim siUnit As String Dim siValue As Double Dim royalValue As Double Dim royalUnit As String Set doc = ActiveDocument Set rng = doc.Content ' Loop through the document and find SI units With rng.Find ClearFormatting .Text = "<[0-9]*[.]?[0-9]*>([m|g|kg|mm|cm|km|ml|l|s|min|h])>" .MatchWildcards = True Do While .Execute ' Get the SI unit and value siUnit = Split(rng.Text, "(")(1) siValue = CDbl(Split(rng.Text, "(")(0)) ' Convert the SI value to Royal value royalValue = ConvertToRoyal(siValue, siUnit) ' Convert the SI unit to Royal unit royalUnit = ConvertUnitToRoyal(siUnit) ' Replace the SI value and unit with Royal value and unit rng.Text = Replace(rng.Text, siValue & "(" & siUnit, royalValue & "(" & royalUnit) Loop End With End Sub
Function ConvertToRoyal(ByVal siValue As Double, ByVal siUnit As String) As Double ' Convert SI value to Royal value Select Case siUnit Case "m" ConvertToRoyal = siValue * 1.09361 ' Convert meters to yards Case "g" ConvertToRoyal = siValue * 0.03527 ' Convert grams to ounces Case "kg" ConvertToRoyal = siValue * 2.20462 ' Convert kilograms to pounds Case "mm" ConvertToRoyal = siValue * 0.03937 ' Convert millimeters to inches Case "cm" ConvertToRoyal = siValue * 0.3937 ' Convert centimeters to inches Case "km" ConvertToRoyal = siValue * 0.62137 ' Convert kilometers to miles Case "ml" ConvertToRoyal = siValue * 0.03381 ' Convert milliliters to fluid ounces Case "l" ConvertToRoyal = siValue * 0.26417 ' Convert liters to gallons Case "s" ConvertToRoyal = siValue * 0.98425 ' Convert seconds to seconds (no conversion) Case "min" ConvertToRoyal = siValue * 0.01667 ' Convert minutes to hours Case "h" ConvertToRoyal = siValue ' Convert hours to hours (no conversion) End Select End Function
Function ConvertUnitToRoyal(ByVal siUnit As String) As String ' Convert SI unit to Royal unit Select Case siUnit Case "m" ConvertUnitToRoyal = "yd" Case "g" ConvertUnitToRoyal = "oz" Case "kg" ConvertUnitToRoyal = "lb" Case "mm" ConvertUnitToRoyal = "in" Case "cm" ConvertUnitToRoyal = "in" Case "km" ConvertUnitToRoyal = "mi" Case "ml" ConvertUnitToRoyal = "fl oz" Case "l" ConvertUnitToRoyal = "gal" Case "s" ConvertUnitToRoyal = "s" Case "min" ConvertUnitToRoyal = "hr" Case "h" ConvertUnitToRoyal = "hr" End Select End Function
` To use this macro, follow these steps: 1. Open your Word document. 2. Press `Alt + F11` to open the VBA editor. 3. Insert a new module by clicking `Insert` > `Module`. 4. Copy and paste the above code into the module. 5. Close the VBA editor. 6. Press `Alt + F8` to open the macro dialog box. 7. Select the `ConvertSiToRoyal` macro and click `Run`. The macro will then search for SI units in the document, convert the values and units to Royal, and replace them in the text.
```

Рис. 9. Код макроса, созданный ИИ, в неструктурированном виде

При доработке кодов не следует игнорировать форумы, на которых обсуждаются вопросы автоматизации операций в текстовом редакторе. Совместными усилиями с участниками форума можно привести код в рабочее состояние без многочасового поиска решения в справочниках, который может и не увенчаться успехом. Пассивное изучение дискуссий также имеет значительный потенциал в расширении знаний в области программирования, углублении понимания потенциала VBA в практике письменного переводчика, результатом чего станут сложные макросы для более глубокой автоматизации профессиональной деятельности.

Выводы

В результате анализа различных аспектов объектно-ориентированного программирования сквозь призму переводческой практики мы пришли к ряду выводов. Во-первых, в сети Интернет уже есть готовые решения для переводчиков на основе макросов. Они ориентированы на преодоление общих трудностей. Тем не менее, данный факт не свидетельствует об отсутствии переводческих проблем, которые можно решить с помощью данного инструмента. Для этого требуется понимание возможностей языка программирования и опыт письменных переводов. Во-вторых, язык объектно-ориентированного программирования, реализованный в пакете MS Office, прост в освоении. Самый оптимальный подход – через практику, а именно – изучение готовых кодов, большинство из которых имеет открытый формат, их последующая доработка и адаптация под частные задачи, компиляция нового макроса из нескольких готовых, автоматическая запись макросов с помощью встроенной функции «запись макроса» с последующим анализом и совершенствованием кода. В-третьих, чат-боты с ИИ также имеют потенциал в создании макросов. Успех решения задачи зависит от качества технического задания, которое невозможно составить без понимания основ объектно-ориентированного языка программирования. Базовые знания понадобятся и на этапе доработки и редактирования кода, который часто не срабатывает из-за незначительных ошибок, допущенных ИИ.

Литература

1. Груздев Д.Ю., Макаренко А.С. Объектно-ориентированный язык программирования и регулярные выражения в практике письменного переводчика // Международный научно-исследовательский журнал «Успехи гуманитарных наук», 2019. № 8. – С. 146–153.
2. Груздев Д.Ю., Макаренко А.С., Шапошник С.В. Методика преподавания дисциплины «Технологии искусственного интеллекта в переводе» // Научно-методический бюллетень Военного университета МО РФ. 2023. № 1 (19). – С. 160–176.
3. Джиноварт К., Светова С. Сравнительный анализ результатов двух опросов о постредактировании МП: европейские и российские заказчики и исполнители. – Режим доступа: https://pemt.ru/wp-content/uploads/2020/09/postredakt_mp_evrop-ross_zakazchiki-ispolniteli.pdf (дата обращения: 10.07.2023).
4. Ельцов К.А. Мастер-класс VBA // Домашнее задание, 1 уровень сложности. – Режим доступа: <https://www.youtube.com/watch?v=q5lJpyRzQGU> (дата обращения: 04.05.2021).
5. Ельцов К.А. Сайт // О технологии E-Stylebook. – Режим доступа: <http://kirilleltsov.narod.ru> (дата обращения: 04.12.2019).

6. Убин И.И. Современные средства автоматизации перевода: надежды, разочарования и реальность // Перевод в современном мире. – М.: ВЦП, 2001. – С. 60–69.
7. Чинаев А.А., Сергеев А.Ю., Филистеев В.В. Инструменты переводчика арабского языка. Свидетельство о государственной регистрации базы данных RU 2021682007. 2021.
8. Шевчук В.Н. Информационные технологии в переводе. Электронные ресурсы переводчика – 2. – М.: Зебра Е; 2013. – 384 с.
9. Choi Y. (2023). Why AI is incredibly smart and shockingly stupid [Video]. TED Conferences. https://www.ted.com/talks/yejin_choi_why_ai_is_incredibly_smart_and_shockingly_stupid/c
10. Ciampi T.H. Internet Search Techniques for Translators. – URL: <http://web.ticino.com/multilingual/internet/internet-course-for-translators.htm> (accessed on 04.05.2021).
11. Gruzdev D.Yu., Dyomochkina V.V., Makarenko A.S. (2021). Leveraging VBA in Translation. Research Result. Applied Linguistics. No. 4. – URL: <https://cyberleninka.ru/article/n/leveraging-vba-in-translation> (accessed on 12.07.2023).
12. Shane J. (2019). The danger of AI is weirder than you think [Video]. TED Conferences. https://www.ted.com/talks/janelle_shane_the_danger_of_ai_is_weirder_than_you_think
13. Город переводчиков: форум // Макросы для переводчиков. – URL: <http://www.trworkshop.net/forum/viewtopic.php?f=16&t=4391> (дата обращения: 04.05.2021).
14. РосНОУ: сайт // Зимняя школа перевода. 2019. – URL: <https://archive.rosnou.ru/important/transschool010219/> (дата обращения: 04.05.2021).
15. DTF: сайт // Российский студент рассказал в Twitter, как защитил диплом, написанный ChatGPT 2023. – URL: <https://dtf.ru/life/1603463-rossiyskiy-student-rasskazal-v-twitter-kak-zashchitil-diplom-napisanny-chatgpt> (дата обращения: 07.07.2023).
16. EnglisHelp: форум // Translator – макрос для переводчика. – URL: <https://www.englishhelp.ru/component/content/article/92.html> (дата обращения: 04.05.2021).
17. GiGatran.ru: форум // Приемы работы с он-лайн словарем «Мультитран». – URL: <https://gigatran.ru/online-multitran-multytran-how-to> (дата обращения: 04.05.2021).
18. Proz: форум. – URL: <https://www.proz.com> (дата обращения: 04.05.2021).
19. Chat-GPT4: AI chatbot. – URL: <https://chat.ai-open.ru/> (accessed on 10.07.2023).

Поступила в редакцию 14 июля 2023 г.

Принята 16 августа 2023 г.

UDC 81'33

DOI: 10.21779/2542-0313-2024-39-1-33–52

Macro-Powered Solutions in Translation

D.Yu. Gruzdev, V.V. Dyomochkina, A.S. Makarenko

Military University; Russia, Moscow, 123001, B. Sadovaya st., 14; gru@inbox.ru

Abstract. The article deliberates some IT aspects in modern translation practice. The researchers note that VBA integrated in MS Office has a potential for fine-tuning its text processing component in line with the needs of modern translators. The obvious obstacle is the requirement of programming skills. The paper results in the scope of problems, which can be solved by ad-hoc macros, recommended approaches to harnessing the instrument, including via AI, and possible application scenarios.

Keywords: code, translation automation, macro, visual basic for application, artificial intelligence, PEMT, measurement unit conversion.

Received 14 July, 2023

Accepted 16 August, 2023