Digital Judaica Done Right

Table of Contents

Introduction

What is this About?

We want to have a computer environment that supports and facilitates study and research of Jewish Orthodox texts. (Most of the technological pieces that we need to develop are not at all specific to Jewish Orthodox Texts. On the other hand, desire to facilitate study of such texts is our primary motivation.)

This "dream system" will record various information about the texts and support doing interesting things with them.

On the text level, we want to allow marking up names of people and places, index entries, logical structure of statements, user-defined tags and the like. On the structural level: allow capturing of multiple structures of the same text, for example - chapter/verse and pages of a printed edition. On the intra-text level: record links from one text to another with their types and other metadata, and support link reversal.

The system should support marking up and working with texts that are stored in different systems (WikiMedia, Sefaria etc.).

It should be possible to print any text with glosses formatted nicely.

The system should be accessible through a clean, even minimalist, but powerful web-based interface. Mass participation - proofreading of the texts, marking them up, clarifying the cross-references and such - should be very much supported.

The system should be also accessible through a web-based API, which the UI should be built on top off (think headless CMS). This API should support modification of the document text as text, but also markup-aware operations. Everything doable via UI should be doable via API.

Texts should be retrievable and modifiable in various formats (primarily - TEI), thus supporting text editor workflow.

API and UI built on top of it should support editing and publishing text like this paper :)

History

Initial conversations in this area took place in 1992-1994, between Leonid Dubinsky and Baruch Gorkin. Most of the requirements were understood then, but not the need for universal web-availability and crowdsourcing: access to the Internet was not what it is now.

We realized that a standard approach to text markup has to be used, and settled on SGML (XML did not yet exist).

In the summer of 2006, discussions restarted between Dubinsky and Gorkin. Crowdsourcing and potential commercialization were discussed. In the Fall of 2006, Michoel Koritz joined in.

In 2019, work on the 19 Kislev archive (www.alter-rebbe.org) lead to a renewed interest in the project.

What do we want? (The Dream)

Texts etc.

We need to store texts in different languages, some of which are written right-to-left. Some Hebrew texts will have vowels, some - cantillation signs, some - special glyphs (small, big, inverted etc.).

We need to store photographs of manuscripts, book scans, possibly - audio and video.

Markup

We need to be able to mark up textual features (encode information contained in the text):

- People names
- Geographical names
- First words of comments
- Authors of statements
- Inference rule used in a fragment of Talmud
- Index entries for a text fragment

We also need to be able to associate metadata with a text fragment:

- What stage of proofreading is the fragment in?
- Tags

Hierarchical Structure

Store hierarchical structure of texts and use it for references and retrieval of text fragments. Examples: Tanach - book, chapter, verse; Chumash - weekly portion; Chumash - parsha (with type - open/closed); Rashi on Tanach: book, chapter, verse, comment; Mishna: treatise, chapter, mishna; Talmud: treatise, folio, side; Talmud - treatise, chapter, statement; Rambam: [book,] laws of, law, statement; Shulchan Aruch: division, chapter, paragraph, small paragraph…

Store page (and line) breaks for multiple editions of the same text.

A text can have multiple hierarchical structures, some of which can be incompatible with one another: parsha can end inside a verse, weekly portion - inside a chapter, page break can be inside a sentence…

Some texts have the same structure, although they are not commentaries on one another, e.g.:

- original and translation
- different editions of the same text
- Shulchan Aruch and Shulchan Aruch HaRav

We need to be able to combine texts with the same structure - e.g., parallel translation. Specific edition gets chosen based on the user preferences: language, presence of vowel points etc.

We need to be able to show differences between different editions of the same text - in a form of a text, with differences highlighted :).

References

Texts reference one another. A reference links point or interval in one text with a point or an interval in another (or the same) text.

References can be external to the texts they link, e.g., parallel statements in Talmud or sources in Shulchan Aruch.

References can have different semantics, which we should store:

- one end comments on the other
- one end proves or illustrates the other
- one end transcribes or translates the other

References can have different "strengths".

References should be reversable: enumerate references that end in a given interval.

Corrections

Correction of one text by another is a specially-handled type of reference.

Texts can correct other texts (Rashi - Talmud) or themselves (Talmud - quotes from early sources).

Text can correct references (from Talmud to Tanach) and structure of another (break up of laws in Rambam).

Attribution

We need to store many versions ("editions") of the same text. This includes typing-in, proofreading and corrections to the text by a user: that's an "edition" too.

We need to develop a theory of attribution for Talmud etc.: "A says in the name of B in the name of C", "two students of B say in accordance to B's views". We should be able to retrieve a text "as seen through the eyes of A".

So: Chumash, Keter edition, according to Peter; ((Rambam through the eyes of Rosh) Romm edition) according to Paul.

Reference to a text that has different "editions" should be resolved in accordance with the user preferences: language, presence of vowels etc.

Versioning

We need to store the history of changes.

Search

A query language provided by the API should allow selection of a subset of texts and support text search that takes structure of texts, markup and grammar into account. e.g:

- by keywords
- all mentions of a city
- all statements by an author
- by language
- latest additions
- by groups of users
- close by the "crowd opinion"
- by "crowd rating"

See Information retrieval from annotated texts by A.S. Fraenkel, S.T. Klein. J.

Individualization

- Personal study program
- Daily study schedule with a list of what you "owe"
- Notebook - selections of text fragments via search of references. Compounding. Storage. Printing.

Crowdsourcing

- Typing in of the texts
- Proofreading: Wikipedia, Wikisource, Distributed Proofreaders
- Marking the texts up
- Adding references
- Annotations
- New presentation styles (XProc/XQuery/XSLT)
- New printing styles

Typesetting

We need to be able to typeset a tree of interlinked texts.

Miscellaneous

- Integration with blogs etc.
- Discussion forums
- Digital libraries
- User levels: guest, registered, editor; "editor, make an editor"; reputation.
- Protection from sabotage: Wikipedia
- Registered domain name opentorah.org.
- Look into publishing an ODPS catalog.

Interface

Передвижение по текстам - горизонтальное и через таги (смысловое); поиск; выбор "фокуса": даф/сугъя; заметки: внести/просмотреть мои; недельная глава, последние и ближайшие шиурим, прошлые поиски юзера, последние поступления и т.д. От текста переход на соседние логические единицы текста, комментарии к нему (к выделенному юзером отрывку), поднятие к

комментируемум им тексту, переводы и варианты. Список просмотренных сегодня текстов. "рабочий стол": выбранные тексты и большой лист для записей юзера - план урока или хидуш (конспект проведенной работы).

Отец семейства хочет подготовить субботний разговор. Мы помним его любимых комментаторов , ему они предложены на "столе", при желании он находит дополнительные материалы на "полке", вытаскивает понравившиеся на лист, возможно добавляет список вопросов для детей. Текст и добавления идут в одном потоке

Подготовка драши к событию. Юзер выбирает из списка (бар мицва, бат мицва, брит, сиюм …) события, затем из другого списка - шиурим ему подходяшие (недельная глава, Тания, Рамбам, ближайшие праздники) и на основе этого выбора он получает набор текстов.

Кроме побора текстов в формате "форума" может понадобиться например снимок листа Гемары.

Для урока в ешиве тихонит учитель может захотеть добавить виде-аудиоматериалы и разные картинки. (При обращени к внешним материалам надо продумать политику цензурирования, чтобы досов не спугнуть)

Презентации.

Web API

Everything doable using the interface should be doable using the Web API.

- Retrieval and modification via various protocols, primarily - HTTP (AtomPub, WebDav, XML-RPC?)
- Retrieval and modification in various formats, primarily - TEI.
- Add/change; add/change metadata.
- It should be possible to work with the text in a text editor.

Attraction and Commercialization

Guilt

Our system must become a part of Jewish culture. A bochur that does not curate a folio of Talmud or a chapter of a rishon will be ostracized. Nobody will deal with a

publisher that did not gift us 10 electronic texts. All sponsors will be ours: we are visible across the world. We will be the place to perform a commandment of writing the Scroll, give haskomos, print hiddushim (like the physicists do in arXive). And to leave a memory of yourself or other people.

Graduated paid services

Additional services for money: quality printing, access to the "super-proofread" texts.

Access based on the purchase of the print book.

Google

They can host and pay for this - but looks like they already did Sefaria :)

Sources

[Fraenkel97] The Responsa storage and retrieval system-whither?. Aviezri Fraenkel. 1997. http://www.wisdom.weizmann.ac.il/~fraenkel/Papers/trs.ps http://www.wisdom.weizmann.ac.il/~fraenkel/Papers/pha.ps

[Ontology] Ontology is overrated. Clay Shirky. 2005. http://www.shirky.com/writings/ontology_overrated.html

[DPR] Distributed Proofreaders. http://www.pgdp.net/c/default.php

[TEI] TEI. http://www.tei-c.org/release/doc/tei-p5-doc/html

[eXist] eXist XML database. http://exist.sourceforge.net

[Stylus] Stylus Studio. http://www.stylusstudio.com

[xmlspy] ALTOVA xmlspy. http://www.altova.com

[oXygen] oXygen. http://www.oxygenxml.com

[editix] editix. http://www.editix.com

[Unicode] Unicode. http://www.unicode.org

[TML] Theological Markup Language. http://www.ccel.org/ThML/ThML1.04.htm

[TanakhML] Tanakh ML. http://tanakhml2.alacartejava.net/cocoon/tanakhml/index.htm

[OSIS] Open Scripture Information Standard. http://en.wikipedia.org/wiki/Open_Scripture_Information_Standard

[NoNew] No new XML languages. http://www.tbray.org/ongoing/When/200x/2006/01/08/No-New-XML-Languages

[nrsi] http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&item_id=XSEM

[Chabad]http://books.chabadlibrary.org/default.aspx

Implementation

Approach

О стандартах

Если есть стандарт, то ясно, что лучше использовать его, чем своё, доморощенное. Выгода от этого понятна: стандарт поддерживается всеми (или многими), а доморощенное - никем; программы, понимающие стандарт, используются широко и отлажены лучше, чем будут отлажены доморощенные (которые ещё и писать придется); люди про стандарт слышали и знают, как с ним работать и т.д. Но главное - сам стандарт, будучи результатом чудовищного количества труда специалистов, как правило "отлажен" лучше, чем любая частная разработка.

Бывает, что стандарт "не прижился". Тогда многие из выгод от его использования пропадают. Но если в какой-то области есть "прижившийся" стандарт, понятно, что игнорировать его очень глупо. Несмотря на то, что из-за "комитетности" разработки многих стандартов в них случаются компромисы, а из-за длительности процесса стандартизации "последнее слово" в них может быть и не отражено.

Тексты на разных языках, справа на лево, с кантиляцией...

Ясно, что тексты должны храниться в Unicode. Придумывать свою кодировку неразумно.

Ясно, что тексты должны храниться в XMLe, несмотря на то, что он не рассчитан на представление нескольких структур одного текста (см. ниже). Тем не менее, придумывать свой, "улучшенный" XML неразумно.

TEI

Один из авторов XMLa, Тим Брай, велит не изобретать своих форматов XMLa, а воспользоваться одним из пяти "основных". В области представления в XMLe "гуманитарных" (извините за выражение) текстов есть стандарт (не включённый Браем в число "основных"): рекомендации TEI (Инициатива

Кодировки Текстов). Долгие годы его разработку возглавлял другой из авторов XMLa - Майкл Сперберг-Маккуин. Ясно, что надо им воспользоваться.

(С другой стороны, хорошо бы понять, почему многие им не пользуются или пользуются лишь частично: Theological Markup Language, TanakhML, Open Scripture, Project Gutenberg.)

Особые буквы

В наших текстах могут быть особые буквы. В TEI вопросами кодировки особых букв занималась специальная рабочая группа. Им посвящена глава рекомендаций.

Аннотации

Аннотации - место, имя ... - в TEI есть.

Перекрывающиеся структуры

Наши тексты могут иметь несколько перекрывающихся иерархических структур. Причем это касается не только Танаха или текстов с многими изданиями и границами страниц. Один из фундаментальных вопросов, на которые должно уметь отвечать наше текстохранилище, это "какие тексты ссылаются на данный". Ответ на такой вопрос видится мне как интересующий нас текст в который добавлены "обратные" ссылки на тексты, на него ссылающиеся. Но "концы" ссылок - которые теперь стали "началами" обратных ссылок - это фрагменты нашего текста, и они запросто могут перекрываться.

Какое-нибудь решение этой проблемы можно придумать не сходя с места. Возможно, даже несколько. Но продумать их во всех деталях, попробовать на практике, сравнить и т.д. займёт годы. Люди, занимающиеся TEI, их уже потратили, уделили этому вопросу главу Рекомендаций, организовали рабочую группу, и продолжают тратить.

Справа на лево XXX програмный интерфейс?

Наши тексты пишутся в основном на иврите, арамейском и идише - справа на лево. Таги TEI (и всех известных мне XML-форматов) пишутся по-английски и, естественно, слева на право. Хорошо известно как представить

двунаправленный документ в XHTMLe так, чтобы все шло в нужную сторону, и чтобы при этом не использовались невидимые символы Unicoda, меняющие направление текста. Нам, однако, надо облегчить редактирование наших текстов в текстовом редакторе (возможно, понимающем XML). Если таги пишутся не в том направлении, что текст, такое редактирование практически, на мой взгляд, нереально. А без использования невидимых символов изменения направления - невозможно.

Упражнение: используя ваш любимый редактор, введите таги посука и , а потом напечатайте между ними посук на иврите. Не столкнулись ли вы с неожиданностями? Например, не меняется ли направление текста когда вы вводите пробел рядом с угловой скобкой обрамляющей таг? Не вводятся ли при этом слова в обратном порядке? К какому слову посука ближе отркывающий таг - к первому или к последнему?

Я не уверен, что если сами таги будут на иврите, то все проблемы ввода текста исчезнут - но я уверен, что хуже не станет. Есть ещё одна причина хотеть, чтобы таги были на иврите: многие наши потребители и участники английского не знают, и даже в пределах набора тагов TEI узнавать его не захотят - и я их понимаю. Было бы неправильно лишить возможности серьёзной обтаговки именно тех, кто на неё больше всех способен. А серьёзная обтаговка возможна только в текстовом редакторе: не только потому, что часто это удобнее, чем всевозможные web-интерфейсы, но и потому, что web-интерфейса, поддерживающего все таги TEI нам не написать. А в серьёзной работе очень многие из них нужны.

Казалось бы, если таги в наших текстах будут на иврите, то это уже не TEI? Не тут то было! TEIвцы начали работать над интернализацией: хотят сделать свою штуковину доступной неанглоязычным. Вообще, у них в последней версии - P5 - пользователь может адаптировать схему, которую генерирует программа ROMA, на свою ситуацию.

В любом случае, мы можем хранить тексты в TEI, но позволять доставать их в другом формате, менять и засовывать обратно. Многие так и делают. Так мы можем, например, ввести структурные таги, более уместные в конкретных текстах, чем довольно общие структурные таги TEI.

Ссылки

Наибольшее беспокойство у меня вызывают ссылки. Они в TEI могут оказаться недостаточно мощными и гибкими. Нам, похоже, просто XLink (XPointer?) не подойдёт: надо посмотреть на Topic Maps и RDF.

Редакторы XMLa

Наш web-интерфейс должен поддерживать довольно серьёзное редактирование документов на XML. Редакторы такого рода существуют. Однако, как ни крути, а надо мочь редактировать наши тексты (XML, TEI) в нормальном редакторе тоже.

Technology

Связаные разработки

Sefaria, OtzarHaHocmo

XML Databases

It is possible to store the texts as XML files in the file system and use XSLT (as implemented by Saxon) to select requested pieces and transform them into presentation form. Indeed, I'll have a copy of all the texts in simple XML files anyway, since I need to check the texts into a revision-control system.

It seems likely, though, that I'll need to store the texts (also) in an XML database. Here are some requirements that make me think so:

- Access parts of documents in response to a query
- Fetch fragments of the documents referenced from a given one
- Find documents referencing a given one (link reversal)
- Full text search

Only first of these requirements can realistically be satisfied without some indexes. On the other hand, only first two are trivially satisfied by an XML database (like Exist). Integration between Lucene text indexing package and Exist needs to be looked into. As for link reversal, we'll probably have to write the indexer and accessor ourselves…

It is clear that a query language to be used is XQuery. It is a nice, functional, non-statically-typed language, that have recently acquired update and text search capabilities. (XXX)

TEIвцы тоже согласны, что надо пользоваться XMLьными базами данных и XQuery [18].

Информацию о различных XMLьных базах данных приводит Bourret.
Некоторые бесплатные базы данных для XMLa:

- eXist
- Berkeley DB XML
- Sedna
- Timber
- MarkLogic
- Lucene

XQuery

Some use XQuery as the (almost) only implementation language for the application (e.g., AtomicWiki). XQuery **is** a functional language. But XQuery does not have static typesystem or exception processing. I will use Scala as my main implementation language, and XQJ to access XQuery/XSLT processors.

XML and Java

There are APIs for

- parsing: javax.xml.parsers
- XSLT: javax.xml.transform
- XPath: javax.xml.xpath
- XQuery (XQJ): java.xml.query

javax.xml.xpath only supports XPath version 1

It seems that I can do pipelines using XQJ.

XML Pipelines

- Cocoon
- Pipelines

- XProc
- Calabash

Пока что я обнаружил только две системы текстохранилищ ориентированных на TEI: Versioning Machine Versioning Machine и . Обе делали одни и те же люди - Susan Schreibman и Amit Kumar, и обе заглохли. Вторая даже использовала eXist.

Нам надо хранить не только сам документ, но и историю его изменений: кто, когда и что. Это даёт возможность вернуться к любому состоянию, посмотреть историю, заблокировать слишком быстрое изменение текста и т.д. Дя этого надо прирастить к базе данных готовую version control system (XXX не очевидно), а именно - GIT.

К текстохранилищу должен быть доступ через сетевые протоколы, а не только через web-интерфейс.

Look Into

AtomPub, WebDav, REST, XML-RPC, XML:DB, GIT, Atom, RSS, Citizendium, Annotea, Collate/Anastasia, XForms

URLs

XPointer in the URI, not in the fragment! No delimiters, just URI parts - which can be implicit (not "chapter=3", but "chapters/3", or just "3")! Editions in the URI ("Chumash/boston+toronto/Genesis")! Metadata ("about"), raw XML etc. - in the URI, not as query parameter ("Genesis/about", "chaters/1/raw")! More URI promotion: natural references ("Genesis/2:1", "Genesis 2:1")! Intervals ("Genesis/2:1-3")! Concatenation ("Genesis/2:1-3;5") probably shouldn’t be done through URIs!

Books URIs:

/books/Tanach/editions/…/[parts/…/]books/…/[weeks/…]/chapters/…/verses/…

editions: a | a+b (side-by-side) | a-b (differences)

parts: Torah | Neviim | Ksuvim

books: Genesis | Ionah | … (appropriate for part if present)

weeks: Genesis | Noah | …

chapters: n | m-n

verses: n | m-n (can be present only if one chapter is selected)

Alternative names may be used.

URL may be truncated.

Parts of the URL may be implied - and need to be derived.

Metadata

Metadata is used to:

- guide navigation
- provide listings and names
- create classifications (links)
- stitch together data directories
- store application-specific metadata

Some of the data in it has to be duplicated in the text document (for self-containment, **and** for non-position-based navigation).

We need to be able to handle things like "Chumash/books/Genesis/weeks" and "Chumash/weeks" with one metadata document…

Locators for the navigational steps can be: - subdirectory/file - element XPATH - milestone XPATH

1) I need to be able to provide a list of selectors (book name/ chapter number/ verse number etc.) on any level.

2) A selector can have multiple names, which I do not want to duplicate (and maintain) in each edition of the text. So, selector names have to be part of the metadata.

3) A text can have multiple structures.They are important for the metadata also. Restructuring of the text is done by XSLT. It seems logical to use the same for the restructuring of the metadata.

It follows that the metadata needs to be processable as XML (and have format similar to the texts). Do I also need it to be processable (in part) as Java objects (using JAXB) - is not clear.

We are going to use milestones [?] to represent multiple structures.

```
<book n="Genesis">
  <chapter n="1">
      <week n="Genesis" milestone="begin"/>
      <paragraph type="open" milestone="begin"/>
      <verse n="1">
      ....
      </verse>
  </chapter>
  ...
  <chapter n="6">
      <verse n="1">
      ...
      <week n="Noach" milestone="begin"/>
      <verse n="..">
      ...
  </chapter>
</book>
```

Tanach Markup

What are the TEI-appropriate tags for Tanach?How do we represent the paragraph in the middle of the verse?

Super-Wiki

Wiki with multiple formats ⇒; function reversal (TEI→;HTML; edit; back)…

Wiki page rename and links correction - if the wiki itself is in an XML database (AtomicWiki) **with** our link-reversal index, wouldn't it be easier? History will be kept by the revision-control system…

Navigation:

- expand/contract viewport
- move viewport
- switch to a different structure preserving focus (from "lesson" to "chapter" in Tanya, for instance)
- switch to a different edition / look around at editions

Notes

crowd-sourcing TEI files

Web-based IDE with WebDAV's versioning

BUGS

Upstream:

- http://sourceforge.net/tracker/index.php?func=detail&aid=2056090&group_id=17691&atid=117691 exist resolve-url
- http://xmlroff.org/ticket/131 xmlroff tables (fixed)

Sebastian:

- File a bug against FO stylesheets (title, table of contents).
- File a bug about reference shape consistency.
- File a bug about use of @name for reference.

Saxon, Tomcat and relative URIs for the stylesheets. XQuery Server Pages (and eXist).

space before a word that has read/write annotations (Psalm 60)

Styles of biblio references.

Google SSO. GData. RSS/Atom - second edition? Hacking…?

Start working on XSLT: Genesis → FO

leningrad-import:

- remove stylesheet link
- add TEI P5 All declaration; namespace(s)
- makaf

XProc

Discussions as text.

Convince CiteULike to make their XHTML really XHTML, or at least - well-formed XML. Better - parse RIS.

Laying out classic Jewish texts

It is natural for a user, after researching with our system, to desire to print selected texts and fragments for personal - or group - study away from a computer. Such printouts are one-use artifacts. It is clear that ability to produce such printouts must be present in the system from the beginning. The question is: how good the typographically does it need to be?

We need to format a tree of texts: main one, commentaries of it, commentaries on commentaries etc. It is known about each piece of commentaries what is it commenting on. All the font metrics are also known: glyph sizes, what is hanging how low and what is sticking up and how high. Result needs to be readable and (is it a separate requirement?) beatyfull.

To format "like in a book", we need to optimize the following contradicting constraints (the list is probably incomplete):

- the page must be fully covered with print
- comment must start on the same page where what it comments on is
- comment must end on the page it started

Koritz says that we do not need to print books, but "leaflets" instead: text with comments that fit on one page. In the "forum format", whatever that means.

Gorkin says that printing "like in the book" of the multi-layered text is extremely challenging typographically, and thus very interesting, but design of the overall interface of the system is even more interesting - and difficult. And more importants. Also, what exactly are the requirements for the printing facility, and what is their order of importance, will become clear only in the process of using the system. So, initially printing needs to be acceptable, but primitive - we do not have resources to do fancy stuff from the beginning.

Dubinsky says that the format that will "grow" from the use of the system, will turn out to be a familiar to us all format "like in the book", or so close to it, that a solution for one will fit the other; that good leaflet is not easier to print than a book; and that ability to print familiar "book-like" format is neccessary for the psychological comfort of the users. But he also agrees that features and interface of the system are more important.

Thus, everybody agrees that initial printing facility will be "primitive". Gorkin does not want to expend any effort to even find out how primitive. Dubinsky would like to see something acceptable. Nothing of the sort has been found so far. XSL-FO [7] is insufficiently expressive for our problem - even version 1.1, it seems.

Beyond Pretty-Printing: Galley Concepts in Document Formatting Combinators

Nonpareil

iText

XSL-FO 2.0 Requirements

Fonts

SBL font is needed for viewing Tanach.

Sources of Free Texts

Jumping points

- OpenSource Judaism
- allhatorah
- Wikipedia Torah database - done
- Wikisource Judaica Bookshelf
- psychomystic links - done - closed access
- Chabad Library
- Sichos Kodesh - empty
- Otzar 770
- hebrewbooks.org
- chabadlibrarybooks.com
- Seforim Online
- Grimoar - Kabbalah
- jewishcontent.org - for PDAs
- Torah Texts
- chassidus.ru - broken
- Halacha Brura
- Digitized Book Repository (JNUL) - broken
- Otzar HaHochma

Texts

- Tanach (Leningrad Codex)
- Mishna
- Targumim
- Midrash Raba
- Midrash Tanhuma
- Yalkut Shimoni
- Ovos DeRabi Noson
- Sefer HaHareidim

Wikisource

and Mechon Mamre

Tanach, Mikraot Gdolot, Targumim, Mishna, Tosefta, Masechtos Ktanos, Mechilta, Sifro, Sifri, Midrash Rabba, Talmud Bavli, Talmud Yerushalmi, Rif, Rambam, Tur, Shulchan Oruch, Kitzur, Oruch HaShulchan, Shulchan Oruch HaRav, Siddur Tora Or

Texts in English

- Babylonian Talmud: Soncino, Rodkinson
- The Guide for the Perplexed
- Shulchan Aruch

Backlinks

▶ index 1