BRILL

RESEARCH DATA JOURNAL FOR THE HUMANITIES AND SOCIAL SCIENCES 9 (2024) 1–13

BRILL OPEN

brill.com/rdj

Text-Fabric Dataset of the Samaritan Pentateuch

Martijn Naaijer | ORCID: 0009-0006-3325-0614
Corresponding author
Faculty of Theology, University of Copenhagen, Copenhagen, Denmark
mna@teol.ku.dk

Christian Canu Højgaard
Fjellhaug International University College, Copenhagen, Denmark
cch@dbi.edu

Stefan Schorch
Faculty of Humanities, The Hebrew University of Jerusalem, Jerusalem, Israel
stefan.schorch@mail.huji.ac.il

Martin Ehrensvärd
Faculty of Theology, University of Copenhagen, Copenhagen, Denmark
meh@teol.ku.dk

Received 7 September 2023 | Revised 31 July 2024 |
Accepted 25 September 2024 | Published online 24 October 2024

Abstract

In this article, the authors present a dataset of the text of the Samaritan Pentateuch with word-level linguistic annotations. The Samaritan Pentateuch is an important early witness of the Pentateuch or Torah. This dataset is based on a transcription generally taken from manuscript Dublin, Chester Beatty Library 751 (Genesis 1:1–Deuteronomy 32:36) and supplemented from manuscript Nablus (Kiryat Luza), Samaritan Synagogue, Garizim 1, where the former manuscript has not preserved the text (Deuteronomy 32:36b–34:10). The dataset is a Text-Fabric dataset. Text-Fabric is a Python package for processing annotated text corpora, which means that the dataset comes with an app, where the text can be inspected and queried using the annotations. It is also easy to

Published with license by Koninklijke Brill BV | DOI:10.1163/24523666-bja10051
 | ISSN: 2452-3666 (online)

perform textual and linguistic research using Python scripts and to make comparisons with other relevant textual datasets with the same annotation conventions.

Keywords

Hebrew – Samaritans – Pentateuch – Bible – linguistics – digital corpus

– Related data set "DT-UCPH/sp" (a Text-Fabric dataset of the Samaritan Pentateuch) with DOI www.doi.org/10.5281/zenodo.7734632 in repository "Zenodo"

1. Introduction

The Samaritans are an ethno-religious group, which has its origins in Ancient Israel (Anderson & Giles, 2012, ch. 2; Kartveit, 2009, ch. 2). They have their own sacred book, generally referred to as the Samaritan Pentateuch. The text of the Samaritan Pentateuch consists of five books (Genesis, Exodus, Leviticus, Numbers, and Deuteronomy) and is transmitted in Samaritan Hebrew (Ben-Ḥayyim, 2000), which is distinct from other traditions of Hebrew.

The text of the Samaritan Pentateuch is very similar to the text of the Pentateuch of the Masoretic Text, which is a Jewish medieval text tradition with much older roots. The latter is a sacred text of Judaism and Christianity. Note that the Pentateuch, or Torah in Hebrew, is only a part of the Jewish Bible, which contains 24 books.

The textual version of the Pentateuch contained in the Samaritan Pentateuch was finalized in the Hasmonean era (2nd–1st centuries BCE). Still, it is based on older versions and retains many of their features. This conclusion is based mainly on a comparison with parallel texts from the Dead Sea Scrolls, but also with the Masoretic Text and with the ancient Greek translation of the Pentateuch called the Septuagint (Schorch, 2015, pp. 18–26; Tov, 2022, pp. 171–172).

2. Research Problem

Ancient Hebrew did not have a uniform or standardized spelling system. Hebrew has a consonantal script which means that the vowels were mostly not written. For example, the name David is mostly written as דוד, which is

DWD in the ETCBC transcription used in this dataset. The pure consonantal spelling can only be found in the earliest Hebrew inscriptions. In Biblical Hebrew, four of the consonants were also used to represent vowels. These are ו (waw, W), י (yod, Y), ה (he, H), and א (aleph, a glottal stop, which does not have an equivalent in the Latin alphabet) and they are called *matres lectionis* or vowel letters. These vowel letters are mainly used for long vowels. In later texts, the name David was often spelled דויד (DWJD in transcription), where the י (J) does not have its consonantal value, but represents the vowel i. There is a general tendency in Biblical Hebrew for later texts to be written with more vowel letters.

Some manuscripts have a strongly increased use of vowel letters (e.g., Tov, 2022, p. 132). On the other hand, a given word might be spelled less often with vowel letters when used in conjunction with prefixes and suffixes (Barr, 1989, pp. 25–31). All in all, various factors seem to influence the use of vowel letters, but overall, their use is inconsistent (Tov, 2021, p. 330).

Various studies have been written on this problem (e.g., Andersen & Forbes, 1986; Barr, 1989), but most of them discuss only one manuscript of the Hebrew Bible. In our project "Artificial Intelligence and Ancient Hebrew Texts", in which we explore the potential of machine learning and statistics in Biblical Studies, the scope is broadened to a variety of manuscripts. Not only do we take the Masoretic text based on the Codex Petropolitanus into account, but also the biblical Dead Sea Scrolls (https://github.com/ETCBC/dss) and the Samaritan Pentateuch. For the latter text, no openly available digital edition was yet at hand, which is why the present dataset has been developed.

3. Collection and Preparation of the Data

The dataset contains transcriptions of the Samaritan Pentateuch (SP), which were taken from manuscript Dublin, Chester Beatty Library 751 (Genesis 1:1–Deuteronomy 32:36) and supplemented from manuscript Nablus (Kiryat Luza), Samaritan Synagogue, Garizim 1 on places where the former manuscript has not preserved the text (Deuteronomy 32:36b–34:10).

The text was provided by Stefan Schorch in Word files. Schorch is the leader of the Samaritanus-project based at Martin-Luther-Universität Halle-Wittenberg. In this project, the editors have developed a comprehensive critical edition of the Samaritan Pentateuch with textual variants from Samaritan texts in Hebrew, Aramaic, and Arabic, and textual parallels from non-Samaritan texts in Hebrew, Greek, Syriac, Latin, and other ancient languages (Schorch 2018–). The present dataset contains the main text of the critical edition.

We added several linguistic features to the text. Annotating a textual dataset is a time-consuming and tedious task. This problem was partly solved by parsing the Hebrew texts morphologically by a machine learning model developed by Martijn Naaijer (https://github.com/etcbc/ssi_morphology; Naaijer et al., 2023). This model was trained on a Masoretic Hebrew dataset. The trained model is able to make predictions for the morphological parsing of 'new' and unseen Hebrew texts. However, Masoretic Hebrew and Samaritan Hebrew are not identical, and there is, therefore, a 'Masoretic bias' in the predictions, which is not always easy to discover, partly because of the lack of vowels in the texts. One of the ways how Samaritan Hebrew differs from Masoretic Hebrew is in the verbal stems (Fassberg, 2001, pp. 246–247; Florentin, 2013; Hornkohl, 2021). For this reason, we have not yet assigned verbal stem annotations to the SP dataset. Also, the Samaritan text differs in several further linguistic phenomena from the Masoretic Text (MT), in terms of morphology, syntax, and lexicon, which need to be interpreted. Additionally, there are cases where SP and MT contain the same text, but the texts have different interpretations.

An example of Masoretic bias in a prediction can be seen in the words ויפח in Genesis 2:7 and ויסגר in Genesis 2:21. These verbs have identical consonantal text in MT and SP, but the verb forms are different. In both cases, MT has a *qal* here, whereas SP has a *hiphil* stem formation. The model was trained on the consonantal text of MT and is therefore inclined to predict that these words in SP are a *qal*, which is a mistake. Based on the vocalization of the text one can recognize that these verbs have different stems. For this, specialist knowledge is needed (see Schorch [2004] for an overview of cases where SP and MT have different interpretations when the consonantal texts are identical). The book shows that this occurs a few times per chapter, which is not very often, but it is important to be aware of this.

All in all, the model is a valuable tool for annotating the text, but with the data that are currently available, human corrections are necessary to develop a high-quality dataset.

The dataset's consistency is tested after every push to the GitHub repo with GitHub Actions in our test suite (https://github.com/DT-UCPH/sp/tree/main/tests).

4. The Dataset

– **DT-UCPH/sp deposited at Zenodo – DOI:** www.doi.org/10.5281/zenodo.7734632
– **Temporal coverage:** 2nd–1st centuries BCE

The dataset is a Text-Fabric dataset (https://annotation.github.io/text-fabric/tf) and follows the annotation conventions of the *Biblia Hebraica Stuttgartensia Amstelodamensis* (BHSA), which is an open dataset of the Masoretic Text of the Hebrew Bible developed by the Eep Talstra Center for Bible and Computer (ETCBC) of Vrije Universiteit Amsterdam over more than 40 years (Roorda, 2018; Van Peursen et al., 2015; see also https://etcbc.github.io/bhsa). The adoption of the ETCBC conventions makes it easy to work with SP and BHSA together.

The BHSA is structured as a graph with various node types, such as word, phrase, clause, verse, chapter, and book. Each node type has its own features. For more details about the BHSA, see Roorda 2018, sections 3 and 4.

The SP dataset has the same graph structure as the BHSA, but presently, it lacks phrases and clauses. These will be added in future versions of the dataset. The node types in the present dataset are sign, word, verse, chapter and book, each corresponding with the meaning they have in the field of Biblical Studies. The dataset contains 114,890 words, and most of the features are word features.

The book of Genesis in SP starts with:

בראשית ברא אלהים את השמים ואת הארץ

This verse means "In the beginning, (when) God created the heaven and the earth". Here, בראשית consists of two words, the preposition ב "in" and the noun ראשית "beginning". The dataset contains some basic textual features. The text in the Hebrew script can be retrieved using the feature `g_cons_utf8`. This feature has a counterpart `g_cons`, which represents the text of a word in ETCBC transcription. The ETCBC transcription of ראשית is `R>CJT`. Another basic word feature is `trailer`, which represents what comes after a word. This can be an empty string or a space. In the example above, ב is attached directly to the next word. Therefore, the trailer is an empty string, whereas ראשית is followed by a space.

There is a range of textual features for the morphemes of words, each of them represented by a utf8 and ETCBC transcription version. The morphemes that are distinguished are:

- `g_lex`, the lexeme part of the word, without prefixes and suffixes
- `g_nme`, nominal ending
- `g_pfm`, preformative
- `g_prs`, pronominal suffix

– g_uvf, univalent final (e.g., *he locale*)
– g_vbe, verbal ending
– g_vbs, verbal stem

For example, the word מאורות "lights" in Genesis 1:15 consists of two morphemes according to this method: the lexeme מאור and the plural marker ות, which is WT in ETCBC transcription. With the features g_lex and g_nme (or g_lex_utf8 and g_nme_utf8) it is possible to retrieve these morphemes distinctly.

Further, lex (lexeme), sp (part of speech), gn (gender), nu (number) and ps (person), vt (verbal tense) are in the dataset. Next to the feature g_prs for the textual realization of pronominal suffixes, the features prs_gn (gender), prs_nu (number), and prs_ps (person) represent details of the pronominal suffix. The feature language represents the language of a word, which has the value 'Hebrew' for every word in the present dataset. A more comprehensive description of the features and their values can be found in the BHSA feature documentation (https://etcbc.github.io/bhsa).

5. Usage

There are two ways in which Text-Fabric can be used with the dataset. The first way is the Text-Fabric Browser, with which the text can be inspected and queried. The other way is to access the data using a Python script. This requires some programming skills, but it gives the opportunity to query, manipulate and flexibly export the data and to use the dataset with other Text-Fabric datasets.

Text-Fabric can be installed with:

```
pip install text-fabric
```

5.1. *Text-Fabric Browser*

After installation, one can run:

```
tf DT-UCPH/sp
```

Now the latest version of the data is downloaded from the GitHub repository (www.github.com/DT-UCPH/sp) and a new tab is opened in the browser (see Figure 1).

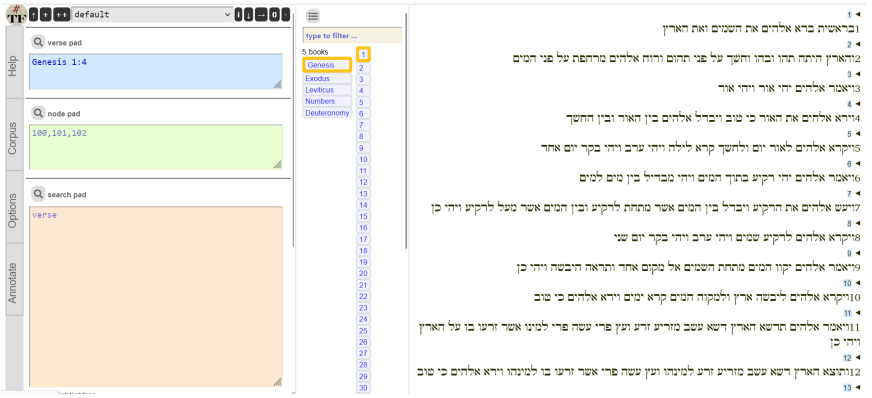

FIGURE 1 Text-Fabric Browser displaying the SP dataset

Here, the text can be explored by clicking on a book name or chapter number. In the search pad, a query can be run using a query language called "Search" (https://annotation.github.io/text-fabric/tf/about/searchusage.html).

With the query `word sp=subs`, Text-Fabric will retrieve all the common nouns in the text after clicking on the magnifying glass. The resulting data can be exported as a tsv file by clicking on "i" and the arrow down (see Figure 2). Note that the query consists of two parts. It starts with `word`, which means that we are searching for word nodes in the dataset with specific characteristics. It is followed by `sp=subs`. `sp` which represents the feature part of speech and, in this case, its value should be `subs`.

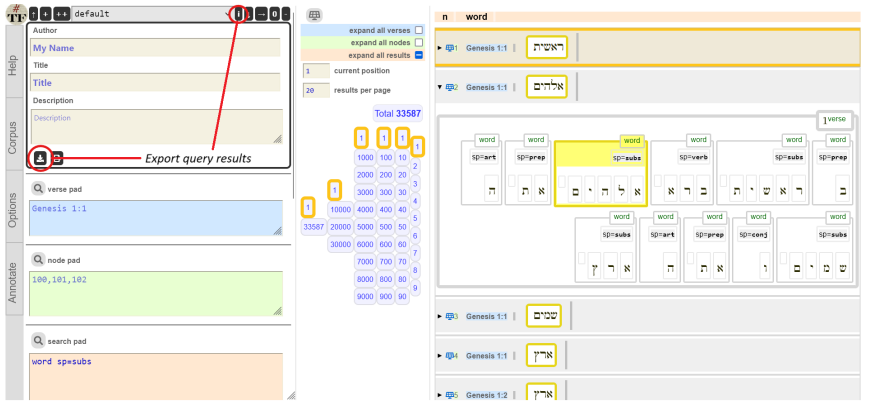

FIGURE 2 Result of a query and export of the data

Extra features can be added to the query. The following query retrieves common nouns having the nominal ending /JM, which usually marks the masculine plural. The result is shown in Figure 3.

```
word sp=subs g_nme=/JM
```

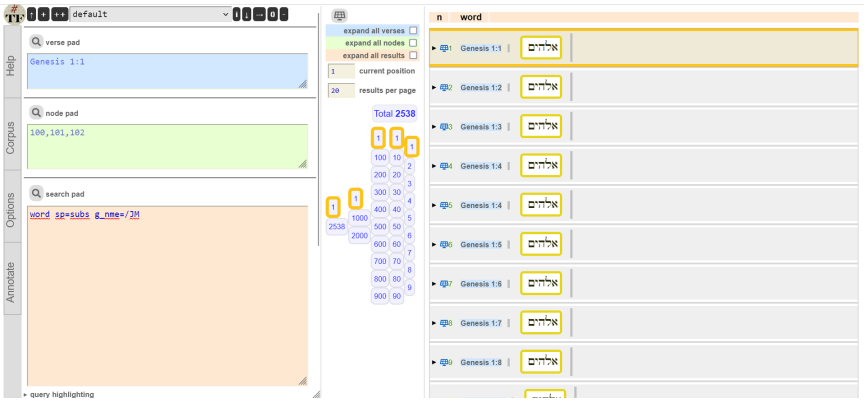

FIGURE 3 Result of a query with two features

5.2. *Python Approach*

The SP dataset can be combined with other Text-Fabric datasets with similar annotation conventions using Python. For instance, if one wants to find out in which verses the consonantal text of SP differs from MT, first both datasets are loaded (see the notebook in Figure 4; the whole notebook can be found here):

In [10]:

```
from tf.app import use

# Load the SP data, and rename the node features class F,
# the locality class L and the text class T,
# then they cannot be overwritten while loading the MT.
SP = use('DT-UCPH/sp', version='3.3')
Fsp, Lsp, Tsp = SP.api.F, SP.api.L, SP.api.T

# Do the same for the MT dataset.
MT = use('etcbc/bhsa', version='2021')
Fmt, Lmt, Tmt = MT.api.F, MT.api.L, MT.api.T
```

Locating corpus resources ...

app: ~/text-fabric-data/github/DT-UCPH/sp/app
data: ~/text-fabric-data/github/DT-UCPH/sp/tf/3.3
Text-Fabric: Text-Fabric API 11.4.16, DT-UCPH/sp/app v3, Search Reference
Data: DT-UCPH - sp 3.3, Character table, Feature docs
Node types

FIGURE 4 Loading SP and MT with Python

Then, the texts of all the verses are reconstructed for both SP and MT using the basic textual features `g_cons` and `trailer`. If the texts are not identical, they are printed (see Figure 5).

In [13]:
```
def reconstruct_pentateuchal_verses(F, L, T, text_feature):
    """For each verse of the Pentateuch in a given dataset, the text of each verse is reconstructed.
    Output:
    verse_texts: dict   Keys are verse label (tuple with book, chapter verse), values are reconstructed text (str).
    """
    verse_texts = {}

    for verse_node in F.otype.s('verse'):
        bo, ch, ve = T.sectionFromNode(verse_node)
        if bo in PENTATEUCH:
            verse_text = ''
            word_nodes = L.d(verse_node, 'word')
            for word_node in word_nodes:
                word_text = eval(f'F.{text_feature}.v(word_node)')
                trailer = F.trailer.v(word_node)
                if not word_text:
                    continue
                elif not trailer:
                    verse_text += word_text
                else:
                    verse_text += word_text + ' '

            verse_texts[(bo, ch, ve)] = verse_text.strip()
    return verse_texts

sp_verses = reconstruct_pentateuchal_verses(Fsp, Lsp, Tsp, 'g_cons')
mt_verses = reconstruct_pentateuchal_verses(Fmt, Lmt, Tmt, 'g_cons')
```

In [14]:
```
for label, mt_verse_text in mt_verses.items():
    sp_verse_text = sp_verses.get(label, '')
    if mt_verse_text != sp_verse_text:
        print(label)
        print('SP:', sp_verse_text)
        print('MT:', mt_verse_text)
        print()
```

```
('Genesis', 1, 11)
SP: WJ>MR >LHJM TDC> H>RY DC> <FB MZRJ< ZR< W<Y PRJ <FH PRJ LMJNW >CR ZR<W BW <L H>RY WJHJ KN
MT: WJ>MR >LHJM TDC> H>RY DC> <FB MZRJ< ZR< <Y PRJ <FH PRJ LMJNW >CR ZR<W BW <L H>RY WJHJ KN

('Genesis', 1, 14)
SP: WJ>MR >LHJM JHJ M>WRWT BRQJ< HCMJM LH>JR <L H>RY WLHBDJL BJN HJWM WBJN HLJLH WHJW L>TWT WLMW<DJM WLJMJM WCNJM
MT: WJ>MR >LHJM JHJ M>RT BRQJ< HCMJM LHBDJL BJN HJWM WBJN HLJLH WHJW L>TT WLMW<DJM WLJMJM WCNJM

('Genesis', 1, 15)
SP: WHJW LM>WRWT BRQJ< HCMJM LH>JR <L H>RY WJHJ KN
MT: WHJW LM>WRT BRQJ< HCMJM LH>JR <L H>RY WJHJ KN
```

FIGURE 5 Text of verses that do not have an identical consonantal text in SP and MT

Suppose one does not want to see the verses with only minimal variation, then it is possible to use the Levenshtein distance to show only those cases in which a minimum distance threshold is exceeded. See Figure 6, where there should be a Levenshtein distance of at least 10:

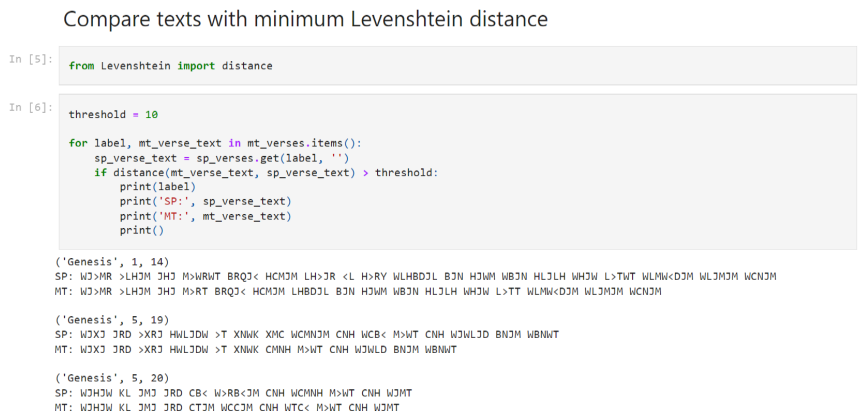

FIGURE 6 Retrieval of non-identical verses with a minimal edit distance of 10

Sometimes, SP has a different spelling of words than MT, where both seem to refer to the same thing. An example is Ararat in Genesis 8:4. It is spelled אררט (>RRV) in MT versus הררט (HRRV) in SP. Other examples are ראומה (R>WMH, "Reumah" in MT) versus רומה (RWMH in SP) in Genesis 22:24 and צחר (YXR, "Tsochar" in MT) versus צהר (YHR in SP) in Genesis 23:8. These spelling differences reflect the weakening of the gutturals in Samaritan Hebrew, which has received an apt description in Ben-Ḥayyim, 2000, pp. 38–43. It must be noted that this variation between MT and SP is not consistent. We can find these cases by searching for lexemes of proper nouns with spelling variations between SP and MT (see Figure 7).

We have chosen to maintain the same lexeme as in MT if the name appears in the same place in the texts of SP and MT. This makes it easier to compare these texts, but we are aware that a different choice could have been made.

For more information about this way of using Text-Fabric, see the documentation (https://annotation.github.io/text-fabric/tf/core/index.html) and a tutorial (https://nbviewer.org/github/etcbc/bhsa/blob/master/tutorial/start.ipynb).

2. Comparison of spelling of proper nouns between SP and MT

In [7]:
```
import collections
```

In [8]:
```
def collect_proper_noun_spellings(F, L, T):
    """Collects different spellings of proper nouns in a dataset.
    Output:
    proper_nouns_spelling: dict  Keys are lexemes of proper nouns, values are set with all spellings of the lexeme.
    """
    proper_nouns_spellings = collections.defaultdict(set)
    for w in F.otype.s('word'):
        bo, _, _ = T.sectionFromNode(w)
        if bo in PENTATEUCH and F.sp.v(w) == 'nmpr':
            proper_nouns_spellings[F.lex.v(w)].add(F.g_cons.v(w))

    return proper_nouns_spellings

sp_spellings = collect_proper_noun_spellings(Fsp, Lsp, Tsp)
mt_spellings = collect_proper_noun_spellings(Fmt, Lmt, Tmt)
```

In [9]:
```
for lex, mt_spelling_set in mt_spellings.items():
    sp_spelling_set = sp_spellings.get(lex, set())
    if mt_spelling_set != sp_spelling_set:
        print(lex)
        print('MT:', mt_spelling_set)
        print('SP:', sp_spelling_set)
        print()
```

```
XDQL/
MT: {'XDQL'}
SP: {'HDQL'}

NWD/
MT: {'NWD'}
SP: {'ND'}

XNWK/
MT: {'XNWK', 'XNK'}
SP: {'XNWK'}
```

FIGURE 7 Spelling variation of proper nouns between SP and MT

6. Conclusions

The dataset offers an easy way to inspect and query the Samaritan Pentateuch. The annotation conventions make it possible to compare its text and features in a simple way with the text of MT and the Dead Sea Scrolls which have the same ETCBC annotation conventions. This facilitates the study of the history and transmission of the text of the Pentateuch.

The annotations are added with the help of a machine-learning model. The predictions of the model are not perfect, therefore we corrected them manually and we will continue to make corrections in the future. We plan to add more word-level features, phrase and clause boundaries and syntax features with the same procedure as we added word-level features.

Parsing done by machine learning has come a long way. But as a stand-alone method, in this case, at least, the results are not completely satisfying. With the addition of human expertise, however, the results are very useful for linguistic analysis. Parsing takes a long time for a human to do, but when not parsing from scratch but rather correcting the output of the machine learning parsing process, an astounding number of human hours are saved.

Apart from making it possible to compare the SP text with other texts, SP is an important witness of the text of the Pentateuch with its own distinct features that is now easily accessible.

References

Andersen, F. I., & Forbes, A. D. (1986). *Spelling in the Hebrew Bible* (Dahood Memorial Lecture). Biblical Institute Press.

Anderson, R. T., & Giles, T. (2012). *The Samaritan Pentateuch, An introduction to its origin, history, and significance for biblical studies*. Society of Biblical Literature.

Barr, J. (1989). *The variable spellings of the Hebrew Bible* (The Schweich Lectures of the British Academy 1986). Oxford University Press.

Ben-Ḥayyim, Z., with the assistance of A. Tal (2000). *A grammar of Samaritan Hebrew, based on the recitation of the law in comparison with the Tiberian and other Jewish traditions*. Magness.

Fassberg, S. E. (2001). The movement from Qal to Pi"el in Hebrew and the disappearance of the Qal Internal Passive. *Hebrew Studies*, *42*, 243–255.

Florentin, M. (2013). Samaritan Hebrew: Biblical. In G. Khan (Ed.) *Encyclopedia of Hebrew language and linguistics online*. Brill. http://dx.doi.org/10.1163/2212-4241_ehll_EHLL_COM_00000280.

Hornkohl, A. D. (2021). Niphalisation in Ancient Hebrew: A perspective from the Samaritan tradition. *Journal for Semitics*, *30*(2) [17 pp.]. www.doi.org/10.25159/2663-6573/9207.

Kartveit, M. (2009). *The Origins of the Samaritans* (Supplements to Vetus Testamentum, 128). Brill.

Naaijer, M., Sikkel, C., Coeckelberghs, M., Attema, J., & Van Peursen, W.Th. (2023), A transformer-based parser for Syriac morphology. *Proceedings of the Ancient Language Processing Workshop associated with RANLP-2023, held in Varna Bulgaria, Sept 8, 2023* (pp. 23–29). www.aclanthology.org/2023.alp-1.3.pdf.

Roorda, D. (2018). Coding the Hebrew Bible. *Research Data Journal for the Humanities and Social Sciences*, *3*(1), 27–41. www.doi.org/10.1163/24523666-01000011.

Schorch, S. (2004). *Die Vokale des Gesetzes, Die samaritanische Lesetradition als Textzeugin der Tora, 1. Das Buch Genesis* (Beihefte zur Zeitschrift für die alttestamentische Wissenschaft, Band 339). Walter de Gruyter.

Schorch, S. (2015). Der Samaritanische Pentateuch in der Geschichte des hebräischen Bibeltextes. *Verkündigung und Forschung*, *60*(1), 18–29. www.doi.org/10.14315/vf-2015-0104.

Schorch, S., in collaboration with E. Burkhardt, U. Hirschfelder, I. Wandrey, & J. Zsengellér (2018–). *The Samaritan Pentateuch: A critical editio maior*. Walter de Gruyter.

Tov, E. (2021). Orthographic practices in the Biblical texts. In S.E. Fassberg (Ed.), *Hebrew texts and language of the Second Temple Period: Proceedings of an Eighth Symposium on the Hebrew of the Dead Sea Scrolls and Ben Sira* (Studies on the Texts of the Desert of Judah, 134). Brill.

Tov, E. (2022). *Textual criticism of the Hebrew Bible* (rev. and exp. 4th ed.). Fortress Press.

Van Peursen, W. T., Sikkel, C., & Roorda, D. (2015). *Hebrew Text Database ETCBC4b*. DANS. www.doi.org/10.17026/dans-z6y-skyh.