Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 948–952
Marseille, 11–16 May 2020

A Multi-Orthography Parallel Corpus of Yiddish Nouns

Jonne Sälevä
Brandeis University
Waltham, MA
jonnesaleva@brandeis.edu

Abstract

Yiddish is a low-resource language belonging to the Germanic language family and written using the Hebrew alphabet. As a language, Yiddish can be considered resource-poor as it lacks both public accessible corpora and a widely-used standard orthography, with various countries and organizations influencing the spellings speakers use. While existing corpora of Yiddish text do exist, they are often only written in a single, potentially non-standard orthography, with no parallel version with standard orthography available. In this work, we introduce the first multi-orthography parallel corpus of Yiddish nouns built by scraping word entries from Wiktionary. We also demonstrate how the corpus can be used to bootstrap a transliteration model using the *Sequitur-G2P* grapheme-to-phoneme conversion toolkit to map between various orthographies. Our trained system achieves error rates between 16.79% and 28.47% on the test set, depending on the orthographies considered. In addition to quantitative analysis, we also conduct qualitative error analysis of the trained system, concluding that non-phonetically spelled Hebrew words are the largest cause of error. We conclude with remarks regarding future work and release the corpus and associated code under a permissive license for the larger community to use.

Keywords: Yiddish, transliteration, diacritization, resource creation

1. Introduction

Yiddish is an example of a resource-poor language, exhibiting both general resource-scarcity and significant orthographic variation. In this paper, we present a multi-orthography parallel Yiddish corpus based on Wiktionary. The corpus contains Yiddish nouns in several orthographic forms, and is intended to serve as a seed for further development of transliteration and orthographic standardization systems. Our contributions are as follows:

1. We scrape all Yiddish nouns[1], and generate a corpus containing parallel versions of each word in romanized, YIVO, and diacriticless Chasidic orthography.
2. We bootstrap a transliteration model using *Sequitur-G2P* (Bisani and Ney, 2008)[2] and obtain results that are sufficiently performant to be used in downstream tasks.
3. Finally, we release the corpus and code under a permissive license. Our hope is that future research will use them to create larger, more unified language resources for Yiddish.

2. Orthographic Challenges of Yiddish

Yiddish has been traditionally been written using a modified version of the Hebrew alphabet (Jacobs, 2005). As a lower-resourced language, Yiddish presents several unique challenges to the development of language resources, and, ultimately, NLP systems.

2.1. Orthographic Variants

In terms of modern usage, there are three main orthographic variants of Yiddish. The first major variant is the standardized Hebrew spellings developed by the *YIVO Institute for Jewish Research*[3], as well as their gold standard romanized forms. The other prominent variants are the Hebrew spellings used by the Orthodox/Chasidic Jewish community, which frequently appear in community publications.[4]

For the most part, the YIVO orthographies have a one-to-one correspondence with the actual pronunciation of the words, which enables relations between the Hebrew and romanized representations to be constructed with relative ease. A significant exception to this are words derived from Hebrew and Aramaic, which retain their original spellings when using the Hebrew alphabet.

Compared to the YIVO standard, the “Chasidic” spellings tend to lack *diacritics* which are used in the YIVO standard to indicate vowels as well as the consonants *b, v, f* and *p*. Consequently, romanizing text written in the Chasidic orthography is much less straightforward than when using YIVO Hebrew spelling as source-side data. Like the YIVO standard, the Chasidic system also retains the original non-phonetic spellings of Hebrew- and Aramaic-derived word forms. Examples of words spelled in the various orthographies can be seen in Tables 1 and 2.

Finally, there are also other less prominent orthographic variants, such as the spelling standard used in the Soviet Union (Erlich, 1973), characterized by the lack of word-final character forms like ך and ף, as well as completely phonetic spelling of Hebrew/Aramaic-derived words, e.g. כאַסענע instead of חתונה. However, as such spellings are nowadays largely obsolete in online Yiddish writing, they have been excluded from the present work.

2.2. Resource Availability

A large hindrance in the development of Yiddish language resources is the lack of suitable, sufficiently processed data sets. Firstly, a lot of resources, such as the digitized e-book

[1] `https://en.wiktionary.org/wiki/Category:Yiddish_nouns`

[2] `https://github.com/sequitur-g2p/sequitur-g2p`

[3] `https://yivo.org`

[4] We summarize the differences between orthographic variants in this section; for a full treatment, see Blum (2015).

Romanized	YIVO	Chasidic
bafelkerung	באַפֿעלקערונג	באפעלקערונג
brudershaft	ברודערשאַפֿט	ברודערשאפט
frumkayt	פֿרומקייַט	פרומקייט
zakh	זאַך	זאך

Table 1: Non-Hebraic/Aramaic word forms in romanized, YIVO, and Chasidic orthography.

Romanized	YIVO	Chasidic
bas-malke	בת־מלכּה	בת־מלכה
khasene	חתונה	חתונה
yeshive	ישיבֿה	ישיבה

Table 2: Hebraic/Aramaic word forms in romanized, YIVO and Chasidic orthography.

collection of the Yiddish Book Center[5] are not available in plain text but only as PDF files. Secondly, when plain text resources do exist, they are often written in a non-standard orthography. For example, the Yiddish Wikipedia[6], is written largely in the Chasidic orthography, along with a small proportion of YIVO orthography mixed in. The use of non-standard spellings makes romanization non-trivial, and creates problems for system development since the Unicode-free romanized forms are often the simplest to work with in the context of building language technology systems. Finally, each collection of text typically only exists in one orthography, which further complicates the task of learning transliteration mappings due to a lack of parallel texts.

3. Related Work

For the most part, previous work on Yiddish NLP and corpus creation has been scarce. Most efforts have attempted to address the resource availability bottleneck, and have focused on the creation of annotated corpora. Examples of such work include the aforementioned Digital Yiddish Library by the Yiddish Book Center, as well as the AHEYM speech corpus (Ćavar et al., 2016). There exists also *The Penn Yiddish Corpus* (Santorini, 1997), which contains romanized Yiddish sentences along with POS tags and syntactic parse trees. Unfortunately, the romanizations are largely *ad hoc*, and do not correspond to the YIVO standard.
In terms of orthographic standardization, Blum (2015) utilized old Yiddish books and Optical Character Recognition outputs to transliterate non-standard Yiddish spellings into standard YIVO spelling. However, it is not clear whether the associated models or corpora have been made publicly available.

On the machine translation front, Genzel et al. (2009) created a English-Yiddish and Yiddish-English machine translation system based on a "convenience sample" of Yiddish data scraped off the internet, and post-processed using various heuristics. While the performance of the translation system itself was promising given the resource-scarcity, less heuristic approaches are desirable to ensure generalizability.

Finally, there has been some work on creating Yiddish word embeddings, as the language has been featured in research that has produced word embeddings for several languages[7] at once (Bojanowski et al., 2017; Grave et al., 2018). However, these approaches typically use Wikipedia as a source of training data, and as most of the Yiddish Wikipedia is based on the Chasidic orthography, the embeddings may perform poorly on downstream tasks if the new data uses a standardized orthography like YIVO. This further underscores the necessity to develop parallel corpora and tools for orthographic standardization, in order to ensure the robustness of downstream learned representations.

4. Corpus Creation

4.1. Data Scraping and Cleaning

We scrape all word forms from the Yiddish nouns category of the English Wiktionary[8] using the **lxml** library in Python. Once the words have been scraped, we filter out rows that contain spaces, as those mostly correspond to multi-word expressions, such as idiomatic expressions. After post-processing, we obtain a final corpus of 2750 word forms. The corpus is available for download at `https://www.jonnesaleva.com/multi-orthography-yiddish-corpus`.

4.2. Filtering out Non-Standard Spellings

Once the words containing spaces have been filtered out, we apply a rule-based system to detect potential non-standard spellings among the words. The handcrafted rules are simple, and correspond roughly to detecting mismatches between the romanization and Hebrew alphabet form of the word.

For instance, a word whose romanization contains *ay* but whose Hebrew script representation contains only ײ instead of ײַ is flagged as potentially non-standard.

As the script identifies the potentially non-standard word forms, the user is presented with the option of choosing one or more of the Hebrew spellings in the corpus to represent the YIVO spelling of the given word.

Notably, this approach sometimes results in multiple spellings being chosen, as the same romanization can correspond to multiple valid YIVO spellings, particularly when the romanized form also corresponds to a Hebrew-derived word. For instance, the corpus contains two valid Hebrew alphabet spellings for *oder*: אָדער and אָדר, corresponding to *or* and the Hebrew month of *Adar*, respectively.

Finally, to ensure that the correct YIVO spelling is recorded, the spellings of Hebrew/Aramaic-derived terms whose spelling is non-phonetic are manually looked up in the *Comprehensive Yiddish-English Dictionary* [9] (Beinfeld and Bochner, 2013). As can be seen in Table 2, the YIVO spelling and diacriticless Chasidic spelling of these terms is not always identical.

[5] `https://www.yiddishbookcenter.org/collections/digital-yiddish-library`

[6] `https://yi.wikipedia.org`

[7] `https://fasttext.cc/docs/en/crawl-vectors.html`

[8] `https://en.wiktionary.org/wiki/Category:Yiddish_nouns`

[9] `https://www.verterbukh.org`

	Strings	String errors	Symbol errors	Insertions	Deletions	Substitutions
YIVO → Rom (train)	2475	33 (1.33%)	64 (0.39%)	3 (0.02%)	31 (0.19%)	30 (0.18%)
YIVO → Rom (test)	274	46 (16.79%)	82 (4.58%)	16 (0.89%)	25 (1.40%)	41 (2.29%)
Rom → YIVO (train)	2420	41 (1.69%)	86 (0.51%)	29 (0.17%)	16 (0.10%)	41 (0.24%)
Rom → YIVO (test)	275	60 (21.82%)	149 (7.88%)	49 (2.59%)	28 (1.48%)	72 (3.81 %)
Chasid → YIVO (train)	2439	61 (2.50%)	73 (0.43%)	5 (0.03%)	19 (0.11%)	49 (0.29%)
Chasid → YIVO (test)	274	78 (28.47%)	89 (4.73%)	10 (0.53%)	12 (0.64%)	67 (3.56%)

Table 3: Results of transliteration experiments.

4.3. De-diacritization

After processing the corpus such that only word forms obeying YIVO spelling rules are retained, we produce the Chasidic orthography word forms by simply removing all diacritics from the YIVO spelling. In normal usage, such as on Yiddish Wikipedia, there is some writer-to-writer variation in the diacriticless spellings; however, we opt to remove all diacritics to produce the most challenging training data.

5. Experimental results

To demonstrate a potential use of the corpus, we train transliteration models to map between YIVO, Chasidic, and romanized orthographies. Specifically, we train models for three separate experimental conditions, corresponding to romanization, de-romanization and diacritization. The scenario of mapping from YIVO to Chasidic orthography – or "de-diacritization"– is trivial to implement deterministically by replacing diacritics in words with the empty string; therefore, a special model is not trained for it.

We train our models using a popular grapheme-to-phoneme conversion toolkit, *Sequitur-G2P*. It should be noted that while the performance of *Sequitur-G2P* is by no means state-of-the-art, it is a good candidate for a baseline model as it works well "off the shelf" without requiring a lot of training data. All models are trained using the default settings, except for supplying necessary flags to indicate UTF-8 encoding. As with the corpus creation code, we release the models and accompanying scripts under the MIT License[10]. Error rates are given in Table 3, along with the sizes of train and test sets (in words). All error counts were computed by *Sequitur-G2P*, with "string errors" referring to the number of words in which any error occurred, and "symbol errors" referring to the total number of errors encountered in the train/test set. Errors are defined as insertions, deletions or substitutions.

5.1. Romanization: YIVO → Romanized

The romanization performance of *Sequitur-G2P* is, by and large, impressive, and the model seems able to capture the regularities of the Yiddish writing system very well. This is the case for both Germanic words –e.g. שמירקעז is mapped correctly to *shmirkez*, and אַרבעטער to *arbeter*– as well as Slavic words, where פּיראָג is mapped correctly to *pirog*.

Errors made by the romanization model largely occur in the case of Hebrew/Aramaic-derived words, as they are far less regular in pronunciation. The model tends to under-insert vowels due to the lack of explicitly indicated vowels, e.g. פּרשה is mapped to *prshe* instead of the correct romanization, *parshe*. The model also tends to insert incorrect vowels, such as predicting *haskale* instead of *haskole* for השׂכּלה. Interestingly, this type of error mimics the errors made by beginning students of Yiddish.[11] A full set of sample romanization outputs can be seen in Table 4.

YIVO	Predicted	Gold standard
שידוך	*shidekh*	*shidekh*
שמאַלץ	*shmalts*	*shmalts*
פּרשה	*prshe*	*parshe*
הלכה	*halke*	*halokhe*
השׂכּלה	*haskale*	*haskole*
שיכּור	*shikor*	*shiker*
חבֿרטע	*khevrte*	*khaverte*

Table 4: Sample output produced by the romanization model.

5.2. YIVO-ization: Romanized → YIVO

In terms of converting from romanized orthography to YIVO standard spelling – or, *YIVO-ization* – the model is again able to capture most of the regularities of Yiddish spelling. It successfully handles both Germanic words, such as *geburt*, mapping it to געבורט, as well as Slavic words like *aparatshik* which it transliterates as אַפּאַראַטשיק. Interestingly, the model is also able to capture regularities such as the sentence-initial *shtumer alef*, א, which appears in case a word starts with a vowel other than *e*. Thus, words like *ikh* are successfully mapped to איך instead of יך.

In terms of Hebraic words, the model appears to incorrectly apply the regular spelling rules it has learned to Hebrew words as well. Thus, for instance, *eytse* gets mapped to אייצע as opposed to the gold standard output, עצה. The model does pick up on some patterns between Hebraic words and their romanized spellings, such as the fact that an ending *-ye* can correspond to ה instead of יע.

Finally, as the model is purely statistical and has not received any rule-based input about what sequences are valid Yiddish, it seems like the model often incorrectly uses the non-final forms of letters in a word-final position. As an example, *shif* gets transliterated into שיפֿ instead of the correct form שיף. More examples can be seen in Table 5.

[10] `http://www.jonnesaleva.com/yi-lrec`

[11] Based on the author's anecdotal evidence.

Romanized	Predicted	Gold standard
shiker	שיקער	שיכּור
seykhl	סייכל	שׂכל
tayve	טייַווע	תּאווה
eytse	אייצע	עצה
shif	שיפֿ	שיף
kirkh	קירכ	קירך
aliye	אַליאה	עליה
shire	שירה	שירה

Table 5: Sample output produced by the YIVO-ization model.

5.3. Diacritization: Chasidic → YIVO

While the diacritization model performs quite well, there are obvious drawbacks to its approach. A particularly severe one is the tendency of the model to only predict the top hypothesis for any input, which implies that if several words have the same diacriticless representation, at most one of them can be diacritized correctly. This is can be seen in Table 6, where it is not possible for the model to recover both אָב and אָבֿ, both of which have the same diacriticless representation, אב.

Chasidic	Predicted	Gold standard
אב	אָב	אָב
אב	אָב	אָבֿ
באפטיסט	באַפֿטיסט	באַפּטיסט
אקס	אַקס	אָקס
קוואטע	קוואַטע	קוואָטע
שיין	שייַן	שיין
תחת	תחת	תּחת

Table 6: Sample output produced by the diacritization model.

6. Discussion and Further Work

Given the corpus and models outlined above, there are several possible avenues for future research. As can be seen in Table 3, error rates tend to be rather high, particularly on unseen test data. While usually alarming, we feel that such overfitting is to be expected here given the small training corpus, and the fact that *Sequitur-G2P* has no information about character sequences that are likely in each orthography *a priori*. As a potential avenue for future research, it would be useful to build more bespoke systems where prior knowledge about likely character transductions is explicitly incorporated into the model.

In addition to domain-based prior knowledge, an overall transliteration system could benefit from unsupervised language models trained on YIVO Hebrew and romanized spellings, which could act as regularizers, and contribute additional information in order to obtain a better estimate about the posterior probability of observing a predicted string given a source string. We feel that this could be particularly useful in transliterating the non-phonetic Hebraic words, and could reduce the amount of necessary training data for the model to perform well.

Lastly, while the *Sequitur*-based models do capture a significant part of the more regular components of Yiddish, overall it seems like they could benefit from N-best output and contextual reasoning to handle the more ambiguous cases, such as Hebrew and Aramaic spellings. If trained on a corpus of sentences, it is plausible that a feature-based approach where surrounding words are taken into account on the source side could substantially inform the transliteration of a given focus word. This could be particularly useful in settings where a Chasidic spelling has multiple potential standardized spellings, but only one is correct given the sentence context.

Overall, it is our hope that the present work will inspire other researchers to develop further tools for Yiddish NLP. Motivations for such research involve not only Yiddish scholarship *per se*, but also the prospect of building working NLP systems for languages that are lower-resourced and non-standardized. Interesting ideas for future research include extending the present work to all Yiddish lemmas on Wiktionary, covering obsolete orthographies like Soviet Yiddish, and building robust command line tools for transliteration. Such transliteration models can be used to normalize the orthography of existing corpora, such as the largely non-standardized Yiddish Wikipedia. Once large standardized corpora become available, they can be used to further learn downstream representations such as word embeddings and build end-to-end trainable models.

7. Acknowledgements

The author would like to thank Constantine Lignos for invaluable discussions and encouragement.

8. Bibliographical References

Bisani, M. and Ney, H. (2008). Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451.

Blum, Y. P. (2015). Techniques for automatic normalization of orthographically variant Yiddish texts. Master's thesis, City University of New York.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ćavar, M., Ćavar, D., Kerler, D.-B., and Quilitzsch, A. (2016). Generating a Yiddish Speech Corpus, Forced Aligner and Basic ASR System for the AHEYM Project. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4688–4693.

Erlich, R. (1973). Politics and linguistics in the standardization of Soviet Yiddish. *East European Jewish Affairs*, 3(1):71–79.

Genzel, D., Macherey, K., and Uszkoreit, J. (2009). Creating a high-quality machine translation system for a low-resource language: Yiddish.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Jacobs, N. G. (2005). *Yiddish: A linguistic introduction*. Cambridge University Press.

Santorini, B. (1997). The Penn Yiddish Corpus. *University of Pennsylvania. For details, contact: beatrice@ babel. ling. upenn. edu*.

9. Language Resource References

Beinfeld, Solon and Bochner, Harry. (2013). *Comprehensive Yiddish-English Dictionary*. Indiana University Press.