ВАРИАТИВНОСТЬ ОРФОГРАФИЙ В ИДИШЕ И ПРОБЛЕМА ИХ АВТОМАТИЧЕСКОЙ ТРАНСЛИТЕРАЦИИ

Кирьянов Д. П. (denkirjanov@gmail.com),
Орехов Б. В. (nevmenandr@gmail.com),
Панова Т. А. (missis.hudson@gmail.com)

Национальный исследовательский университет «Высшая школа экономики», Москва, Россия

Ключевые слова: транслитерация, обработка текста, идиш, система на основе правил, орфографическая вариативность, стандартизация орфографии

YIDDISH ORTHOGRAPHIES VARIETY AND PROBLEMS OF AUTOMATIC TRANSLITERATION[1]

Kirjanov D. P. (denkirjanov@gmail.com),
Orehov B. V. (nevmenandr@gmail.com),
Panova T. A. (missis.hudson@gmail.com)

National Research University Higher School of Economics, Moscow, Russia

This study is dedicated to the problem of automatic transliteration of different Yiddish orthographies. Almost every publishing house has its own specific orthographical features and each orthography can be inconsistent. The team of the Yiddish corpus needs a tool that would standardize the variety of the writing systems. There are several types of converters but they can not meet all our needs. The converter that we created works in two steps: firstly, using the complicated rule-based system, it converts any given Yiddish text into standard orthography, secondly, it converts a text in standard Yiddish into one in Latin letters. The units engaged into our rule-based system are mostly morphemes although we used also some other letter combination that ought to be transliterated in a complicated way. Our solutions led to the accuracy of transliteration 94% of raw text and

[1] This study (research grant No 14-05-0074) was supported by The National Research University—Higher School of Economics' Academic Fund Program in 2014

98% of the text written in more or less standard orthography. We think its efficiency can be improved by adding a list of words of semitic origin and by methods of machine learning.

Key words: transliteration, parsing, Yiddish, rule-based system, orthographical variety, orthography normalization

1. Introduction

By now the Corpus of Modern Yiddish (http://web-corpora.net/YNC/search/) has about 4 million tokens, but it consists of modern press mostly, there is a lack of texts composed by the classics of Yiddish literature (such as Sholem Aleichem, Sholem Ash and many others). This situation is caused by a number of reasons. The first of these factors is the difference of orthographies: each of the Yiddish authors has their own while the standard one was specially developed by YIVO (the Institute for Jewish Research), thus we can say it follows YIVO-rules. This standard is a construction of different dialectal features and before the rules were created in 1938 there were no texts written in this standard variant [Fishman].

However, there is quite a widespread point of view according to which it is possible to distinguish some groups of orthographies clearly and exhaustively (see, for instance, [Jacobs 2009: 285–293]). Firstly, Eastern Yiddish (Western Yiddish is out of our consideration here) has three main dialects: northern (Lithuanian), central (Polish) and southern (Ukrainian) and each of them has its own specific orthographic variant. Adding some variants that exist in contemporary areas with dense Yiddish-speaking population (such as some neighbourhoods in Brooklyn and Jerusalem) to the dialectal variation one may conclude that such a list would be exhaustive: there are 5–7 variants of orthographies. However, the situation is much more complicated.

Every publishing house, every period, every area has its own specific orthographical features. We analysed about 50 texts published in different places and time periods and we have found out that we can virtually split them into several groups according to their orthographical features, but these groups are “inconsequent”: it means that it would be wrong to claim that there are groups with strict borders. We can say instead that each group includes a whole set of different variants and there are some points where one can not predict or explain which grapheme will be chosen. Let us consider one example, the word ייִדישקייַט [yidishkayt] ‘Jewishness, Yiddishness’. We attested different variants in books edited by various publishing houses: אידישקייט [idishkeyt] (Melukhe-farlag, Moscow, 1959), ייִדישקייט [yidishkeyt] (Kooperativer folks farlag, New-York, 1938). There are dozens of such points in which the texts can vary.

The case of Yiddish is complicated even compared to languages with multiple orthography standards. For example, in Modern Eastern Armenian there are only two standard orthographies, Classical and so-called Soviet (official in present-day Armenia), so multiple standards were not as much of a problem for the Eastern Armenian National Corpus; besides, texts in the Classical orthography comprise just 3.24% of the entire corpus.

This means that if one wants to create a normalizer in order to include the classics' texts into the corpus one has to compile a list of variants for each grapheme (and maybe morpheme, see the section 4) regardless of the orthographic variant this text follows. The parser needs normalizer in order to assign to each token morphological annotation (the text does not have to be changed here). Nevertheless, there is a positive aspect in the text representation in a unified orthography: it makes the search much easier. Thus it makes sense to use the normalizer in the search engine of the corpus in order to let the user search a word in standard orthography and get both the standard and the original form of the word in the output.

The key point of this methodology is to deduce all the possible variants of a unit. The other steps of our methodology are described below, in the section 3. Besides this, we need to improve the work of the converter implemented into the corpus.

2. Converters that already exist

There are several types of converters that we analysed and whose advantages and disadvantages we should take into account.

The first type includes converters, that transliterate a Yiddish text printed in Hebrew script into Latin script. The most popular converter of this type is "Shrayberke" (http://www.cs.uky.edu/~raphael/yiddish/makeyiddish.html). The input text here can be a plain text typed in a window at the web-page. A user can also upload a file in one of the following encodings: QText, MS-Windows Hebrew, Mac Hebrew, Unicode Unicode-16. The input orthography should be in YIVO standard. This converter is designed not only for Yiddish, but also for Ladino texts in romanized orthography. The program works as a spellchecker and as a converter. The spelling check corrects misspellings according to the YIVO standard and produces hypothesis about the possible correct spelling. The program corrects the mistakes according to words from dictionary of A. Harkavy ([Harkavy 1928]) with some simple morphemes (e.g., a misspelled word with one morpheme will be corrected, while a misspelled word word with several morphemes won't.) Such a correction is not a normalization and some regular spellings in other orthographies will not be corrected. The user can choose the form of the output text: transliteration, IPA or one of two Yiddish orthographies in Hebrew letters (YIVO or Algemayner Zhurnal standard.) Another disadvantage of this converter is as follows: many words of Hebrew origin are absent in its dictionary. The accuracy of the transliteration from the Hebrew script into the Latin script for texts with non-standard orthography does not exceed 80%, while a text in YIVO orthography is transliterated with accuracy circa 99%, depending on the amount of complicated Semitic loanwords.

Another popular tool for transliterating is application "Yiddish Typer" by Google (https://chrome.google.com/webstore/detail/yiddish-typer/higlebidpnhccabnhankdoiiccnnfkko). This tool converts a Yiddish text typed in Latin script into the Hebrew script online. Although this tool does not transliterate Hebrew letters into Latin ones and transliterates the words with Hebrew origin only phonetically (which means that it is not following the YIVO rules to the full extent), "Yiddish Typer" transliterates

Latin script into Hebrew, as the "Shrayberke" does, also according to the rules of "Algemayner Zhurnal". Its accuracy is about 98%, as it does not transliterate words of Semitic origin correctly (as there is no dictionary included in the module).

Finally, there is a converter implemented into the Corpus of Modern Yiddish. It can convert any text given in Hebrew script into the Latin one, but it has no function which could normalize the texts so it is impossible for parser to parse the texts written in non-standard orthography. There is a module in the parser that converts the Hebrew script (YIVO) into the Latin script, and then the parser can analyse the text (so it can analyse the texts given either in Hebrew script or in Latin one). In order to make it possible for the parser to work with texts in non-standard orthography we need either to rebuilt the parser for all the variants of orthography or to transliterate the texts into the standard orthography. We chose the second option that will allow us to add many new texts which are now inaccessible for the corpus. If we normalize the orthography of a text in Yiddish, then it can be uploaded into the corpus. On the technology and requirements to a text which is going to be added into a corpus see [Ljashevskaja et al. 2004] and [Poljakov 2005].

As for normalizers, we do not know any program which could normalize text in non-standard Yiddish orthography.

As a result we need a new converter which would meet the following criteria:

1) it could normalize the text into the standard orthography;
2) it should work with all the possible range of Yiddish orthographies and their features: if a text in the input is given in non-standard orthography, the program ought to be able not to correct the characteristics of the orthography (by the means of spellchecker or any other normalizer) so the text in the output is just the transliteration of unnormalized text in the input;
3) it should have maximal efficiency;
4) it should allow all users to add new items into the dictionary of the words with semitic origin.

When we get such a converter it would help us to add texts written in non-standard orthographies into the corpus because the parser will be able to parse such a text for this text is normalised.

3. Methodology

In this section we will describe our methodology. Usually in computational linguistics transliteration is a topic associated with proper names and technical terms only [Knight et al. 1998], [Haizhou et al. 2004], [Virga & Khudanpur, 2003], [Al-Onaizan & Knight, 2002]. But if we turn to Yiddish we have to apply also some rules of normalization so we should have an option to work in two stages.

We classified all the variants of the letters into rows of letters in Hebrew script, each having one representative letter (in the standard orthography developed by YIVO Institute for Jewish research). Some letters in some orthographical traditions (we took into account about 50 of them) have different positional variants. Some different letters may look similar in some orthographies. For instance, letter ן (n) at the end of the

word may have variant ען (en) in some traditions, thus they have different indexes in our metatable of symbols. Having representative letters we feel free to add new variants to the rows of letters. Each representative letter has only one variant in Latin script. The letters, letter combinations and their variants are presented in a table, each line of which contains a row of variants, and each column stands for the orthographic tradition.

There are two steps in the work of our converter:

1. The program converts the non-standard variant of the orthography in Hebrew script into the standard Yiddish orthography. In the output there is a text in standard Yiddish orthography in Hebrew script. This is the first tool for converting a text into the standard orthography. The program works with all the texts, both in standard and in non-standard orthographies, as even a text in the standard orthography may have mistakes that the converter can correct.
2. The second step is transliteration. This module can work either independently from the first stage described above or after it depending on what user wants to get transliterated in Latin (or other) letters: the original text or the normalized one. The list of systems which can work at input and output stages can increase.

There are words that can not be transliterated automatically, as they are not spelled phonetically. These words have semitic origin and they have consonant script, so the only way to transliterate them correctly is to make a list of such words. This segment constitutes about 2% of the Yiddish vocabulary, and the module which will contain the list of them will be added to the program later.

4. Some complicated cases

During our analytical work with the table described above we faced some difficulties. Some letters should be transcribed in different ways, depending on the context and their position in the word. There are two types of such problematic cases:

1. Some letter combinations that should be transcribed in a specific way that does not depend on the context, for instance, two letters "u" ("וו") always mean "v" and letters "u" and "i" together ("וי") always mean "oy". Such letters as "o" ("אָ") consist of two elements, the second of which is the diacritic: "א" = "אָ" + "ָ". Thus we have to transcribe such letter combinations first.

2. Some letter combinations, that constitute morphemes, should be transcribed according to their position in the word and to the context. These cases are analysed below.

For example, the suffix *heyt* ("הייט") being written in the end of the word before several other morphemes should be first normalized into "הײַט". Then we have to transliterate a combination of two symbols "יי" + "ַ" (*ey* + diacritics) and we will have a partially transliterated word "העyט", that will be finally transliterated into *hayt* according to the unambiguous rules of transliterating single letters.

Another example is the prefix -פֿאַר *far-*: it can be written as -פֿער *fer-* not because the letter alef (*a*) can be written as *e* but because this *morpheme* has various spellings in different orthographical traditions. That is why we added a number of prefixes and

suffixes, but not all of the Yiddish morphemes: for example, the prefix אריַנ *arayn* can be written as אַריַנ *areyn*, but the spelling variant correlates with the "simple" rules of letters transformation.

This way we got 59 rows of letters and letter combinations in our table. Afterwards we are dealing with the question: what should we teach our converter? Do we need, for instance, to teach it morphology? If so, then to what extent? Do we need part-of-speech tagging? In order to answer these and other similar questions we had to study each morphemic case in depth and decide which degree of knowledge of the Yiddish grammar will be enough to cope with this. Below we will analyse some cases:

1) The adjectiviser/adverbialiser -יק [-ik] (can be written as -יג [-*ig*]). In this case (like in most of the others) we can apply only positional rules: if -יג is the end of a word we ought to change it by -יק. However, it is not the final solution because adjectives have some grammatical categories and therefore different endings. Adverbs have comparison degrees. So by this step our rule is: "change -יג to -יק if the left context of -יג are some letters and the right context is space or a set of endings". However, there are some proper names and other nouns which do end in -יג in standard Yiddish, for instance, the word וויג [wig] 'cradle'. That's why the final version of this rule is as follows: "change -יג to -יק if the left context of -יג are some letters and the right context is space or a set of endings and if this word does not belong to a special list of exceptions".
2) There are also more complicated cases. For instance, that of the diminutive suffixes. In Yiddish there are two diminutive suffixes: -ל [-l] (in singular, with the plural ending -לעך [-lekh]) and- עלע [-ele] (in singular, with the plural ending -עלעך [-elekh]). The situation is not so simple because -ל is a syllable-forming sonant and some orthography traditions reflect this fact in the spelling of this suffix, writing it like -על [-el]. However, in standard Yiddish there is a rule according to which the suffix *-l* being used after some letters ought to be written as -על. That is why there is a problem: how to distinguish these cases? If we spell ע before ל, then the plural forms coincide. How to distinguish the plural form? Our solution contains several steps. First, if there is an '-על' we should check, if it is followed by an '-ע'. If so, then we should not change it in any way because this is the second diminutive. If, furthermore, there is -עלעך, we also should not change this although it is not clear which of the diminutives we came across. The speakers can not produce the singular form if the plural form has -עלעך ending. We allow our converter to make the same mistakes people can make. If there is no -עלע, we should check if this word belongs to the list of exceptions (i.e. to those words which end in -על but are not diminutives, e,g. פֿויגל [foygl] 'bird'). If this word does not belong to this list and the last letter before -על is not in the list of letters which force the appearance of -ע before -ל in standard Yiddish, then we should change -על to -ל.

After a deep analysis of all the cases we had in our table we found out that for the beginning we do not need any morphological rules and we can solve all the ambiguous cases according to the position of the letter. Now we do not have to apply morphology, syntax and other language grammar, but only lists of exceptions to the rules. Later, we will have to attach a dictionary to solve ambiguous cases within the stems.

5. Some preliminary conclusions and outlook

The texts we work with are of different quality: some orthographies are close to the standard one, some others are pretty different from this. Therefore the normalization accuracy varies. We checked one text in the standard orthography and two texts in non-standard ones. For the text in standard orthography the accuracy of normalization is close to 100%, as there is nothing to change. The accuracy of transliteration is 2% less, as 2% of the text that were not transliterated correctly are the words of semitic origin that can not be transliterated automatically yet. Every word of semitic origin should be counted as one element of the text equivalent to one letter. These words and their morphology could be completely different from common Yiddish words, so they must be treated as a special case. The normalization accuracy of a raw text, whose orthography is not standard, is about 94–97% depending on the orthography. That means that there are still some cases in the raw text orthography that are converted incorrectly. We see several problems that should be solved to improve our converter.

1. In some orthographies there are "silent" letters, that are not pronounced, but are just borrowed from the German orthography, where they mean that the previous vowel is long. As far as there are no long vowels in Yiddish they must be erased from the word. For instance, the letter *hey* (ה) after vowels usually has no equivalence in both standard Yiddish and transcription, so if the word is קוה *kuh* in the text, it should be converted into קו and transcribed like *ku*. But that is not always so. Firstly, there are many exceptions from this rule, and secondly, sometimes the letter combination עה *eh* means the diphthong *ey*, rather than a single vowel.

2. Another problem that gives a big portion of the mistakes in the work of the converter is as follows. The mistakes that occur within the stem not always can be solved automatically. Some cases do not depend on the orthographical tradition or on the position of the word. The correct writing of such minimal pairs as *meyn-mayn* (מײַן-מיין, "(I) think"-"my") can not be chosen according to our criteria used for all the other cases. There are two possible ways to solve this problem:

1) To check all the ambiguous words in the dictionary;

2) To use the hidden Markov models.

Dictionary can not guarantee a complete coverage of all cases. Language is always wider than any dictionary and "live" texts always have words that dictionary cannot "catch". Of course, they can be added to the dictionary (that is why the dictionary should be user-modifiable). But it is still not a perfect solution. So the prospect of working on the converter is engineering a module that uses the principles of machine learning. We hope it will allow to identify the correct variant for complex cases.

Currently our converter works with a text in any Yiddish orthography as an input and a user can get two entities as an output: first, the text in standard Yiddish orthography, second, the input text (either in its orthography or in the standard one) in US ASCII. However, in the future we will need to extend the functionality of the program and connect other graphics (Cyrillic, IPA etc.) to it, as well as make the converter work both ways. For this case we will use a special layer of prototype Yiddish graphics

which would not have any positional ambiguities. It is the layer through which different graphic systems will connect. It will allow us to list only one set of matches when adding a new graphic system (prototype ↔ new system) instead of matching it with each of the orthographies which are already in the system.

References

1. *Al-Onaizan Y., Knight K.* (2002), Machine transliteration of names in Arabic text, SEMITIC '02 Proceedings of the ACL-02 workshop on Computational approaches to semitic languages, pp. 1–13.
2. *Fishman J. A.* Planning and Standardization of Yiddish. The YIVO encyclopedia of Jews in Eastern Europe, available at http://www.yivoencyclopedia.org/article.aspx/Language/Planning_and_Standardization_of_Yiddish).
3. *Haizhou L., Min Zh., Jian S.* (2004), A joint source-channel model for machine transliteration, ACL '04 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, available at http://delivery.acm.org/10.1145/1220000/1218976/p159-haizhou.pdf?id=1218976.
4. *Harkavy A.* (1928) Yiddish-English-Hebrew Dictionary, NY, Hebrew Publishing Company.
5. *Jacobs N. G.* (2009), Yiddish: a Linguistic Introduction, UK, Cambridge University Press.
6. *Knight K., Graehl J.* (1998), Machine transliteration, Computational Linguistics, Volume 24, Issue 4, December. pp. 599–612.
7. *Ljashevskaja O. N., Plungjan V. A., Poljakov A. E., Savchuk S. O., Sichinava D. V.* (2004), Text processing for the Russian National Corpora: technological chain [Obrabotka tekstov dlja Natsional'nogo korpusa russkogo jazyka], The abstracts of the papers presented at the international conference "Corpus linguistics-2004" ["Tezisy dokladov mezhdunarodnoj konferencii "Korpusnaja lingvistika-2004"], SPbSU, Saint Petersburg, pp. 54–56.
8. *Poljakov A. E.* (2005), The technology of the preparation of information for the Russian National Corpora [Tehnologija podgotovki informacii v Nacional'nom korpuse russkogo jazyka], Russian National Corpora: 2003–2005. Results and perspectives [Nacional'nyj korpus russkogo jazyka: 2003–2005. Rezul'taty i perspektivy], Moscow, pp. 175–192.
9. *Virga P., Khudanpur S.* (2003), Transliteration of proper names in cross-lingual information retrieval, MultiNER '03 Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition, Volume 15. pp. 57–64