Mainstreaming August Strindberg with Text Normalization

Adam Ek
Department of Linguistics
Stockholm University
adek2204@student.su.se

Sofia Knuutinen
Department of Linguistics
Stockholm University
irkn8107@student.su.se

Abstract

This article explores the application of text normalization methods based on Levenshtein distance and Statistical Machine Translation to the literary genre, specifically on the collected works of August Strindberg. The goal is to normalize archaic spellings to modern day spelling. The study finds evidence of success in text normalization, and explores some problems and improvements to the process of analysing mid-19th to early 20th century Swedish texts. This article is part of an ongoing project at Stockholm University which aims to create a corpus and web-friendly texts from Strindsberg's collected works.

1 Introduction

The purpose of the current study is to assert how well the Levenshtein and Statistical Machine Translation (SMT) methods developed by Eva Pettersson work on the 19th century texts of August Strindberg. This is done to see preliminary results on how well the normalization works on the text and in the future apply other Natural Language Processing (NLP) tools to the normalized data.

Normalizing historical text is a fairly new area in NLP, but it has a growing interest among researchers. The main interest has been in normalizing spelling and improving part of speech (POS) tagging. There is earlier research on normalizing historical spelling to modern spelling and to follow are a few examples on some. Rule-based approaches have been studied by Rayson et al. (2005), Baron and Rayson (2008) and Pettersson et al. (2012).

A rule-based study was done by Rayson et al. (2005) on 16th to 19th century English texts. This study resulted in a tool called VARD (VARiant Detector). The VARD tool was compared to the results of modern spellcheckers (MS-Word and Aspell) that are not trained for historical data. The results showed that VARD was more accurate in normalizing historical text. This same tool was later developed even further by Baron and Rayson (2008). They implemented phonetic matching against a modern dictionary and candidate replacement rules for spelling variants within a text.

Pettersson et al. (2012) hand-crafted a small set of normalization rules based on 17th century court records and church texts to improve tagging and parsing of historical text. Their results showed that a small set of rules can work well in normalizing historical text and that it works for spellings even more archaic than their training data.

Pettersson et al. (2013b) researched a Levenshtein based approach which was tested on the same data as Pettersson et al. (2012) used. The Levenshtein approach uses a weighted distance measure for comparing the original word form to word forms listed in a modern corpus or dictionary. The accuracy baseline for the Levenshtein method turned out to become 77% and by adding an error handling cache, in the form of a manually normalized training corpus the results increased even more up to 86,9%.

Pettersson et al. (2013a) approached the normalization task as a form of translation. It resulted in a statistical machine translation (SMT) based approach. The translation was done based on characters and not on words as whole. The results showed that having even a small amount of training data works well and the SMT normalization showed an accuracy increase and reached a level of 86,1%.

This paper deals with spelling normalization of August Strindbergs books, which were written between 1869 and 1909. Strindbergs books are mainly written during the era of Late Mod-

Proceedings of the 21st Nordic Conference of Computational Linguistics, pages 266–270,
Gothenburg, Sweden, 23-24 May 2017. ©2017 Linköping University Electronic Press

ern Swedish. This means that they are a bit more modern than the data Pettersson et al. (2012) used when they began to research the different approaches to text normalization. There is even some shifts in spelling in the books since Strindbergs last books were written at the time when Contemporary Swedish was making its way forward. Strindberg is even considered as one of the authors who had a big influence on the changes of the Swedish language.

The approaches used are Levenshtein-based normalisation and SMT-based normalization which is two out of four possible approaches presented by Pettersson (2016a). It will concentrate on a qualitative analysis on both distinctive features of archaic Swedish and how well they are translated and on a few problematic areas in normalization picked from the data collected.

2 Method

2.1 Data

The data consists of August Strindbergs collected works,[1] of which 59 books have been parsed to XML and then into raw text format. The collected works consists of novels, short stories, poems and plays. The data used for testing is the chapters from each book excluding foreword, afterword, word clarifications, public reception. The data was preprocessed by removing all non-alphabetic characters and tokenizing the text with one token per line.

It is noteworthy that Strindberg uses *code-switching* frequently in the text, which are words that should not be considered as candidates for normalization. Chapters containing only french text has been removed manually and as many individual words as possible have been removed manually by identifying french spelling patterns.

The Swedish corpus, training and tuning data for both methods was provided by Eva Pettersson (Pettersson 2016b).[2] The corpus data (language model data) consists of 1166539 tokens, the training data consists of 28777 tokens and the tuning data of 2650 tokens. The data is from a different genre, court records and church documents, while Strindbergs works are mainly literary fiction. The data is also older (15th to 17th century) compared to the texts of Strindberg which were written from 1869 to 1909.[3] This results in that the training data is not fully optimized for the current task.

The modern language dictionary used for both methods is the SALDO dictionary (version 2) (Baron et al., 2008). The corpus of original spelling mapped to manually normalized word forms is compiled from the Gender and Work corpus of court records and church texts from 15th to 17th century. (Pettersson, 2016b)

2.2 Normalization

2.2.1 Levenshtein

The Levenshtein-based approach calculates the extended Levenshtein edit distance [4] between the original word and any token present in a modern language dictionary. It picks out the candidate(s) with the smallest distance and then proceeds to choose the best candidate out of several (given there are more than one candidate for the word). Perl was used to perform the Levenshtein-based normalization. The data used for this was a modern language dictionary, a corpus of original words mapped to manually normalized spelling, a corpus of modern language to choose the most frequent candidate in the case of more than one candidate, a file containing weights lower than 1 for commonly occurring edits observed in training data and a file containing the edit distance threshold value. All these, except for the input file, were provided in Pettersson's HistNorm package.

2.2.2 SMT

When using SMT, the task of text normalization is seen as a translation, where the translation is between spelling conventions rather than between different languages. This is achieved by using character-based translations, where the individual characters rather than words are treated as the lowest level entity, words are treated as middle level, and sentences as the top level. The SMT-based normalization was performed using Moses Statistical Machine Translation System,[5] GIZA++[6] and SRILM.[7]

[1] 72 books in total provided by Litteraturbanken.

[2] http://stp.lingfil.uu.se/~evapet/histNorm/

[3] Strindberg wrote his later texts just as the Swedish spelling reform of 1906 took place. It is unclear if he decided to follow their directions.

[4] Extended edit distance uses both single character edits and additional operations such as double deletion and single-to-double substitution (Pettersson,2016a)

[5] http://www.statmt.org/moses/

[6] http://www.statmt.org/moses/giza/GIZA++.html

[7] http://www.speech.sri.com/projects/srilm/

The input data was tokenized, then each character in the word was separated with a whitespace to indicate that each character should be regarded as the lowest level entity, and newlines to indicate the end of a paragraph.

In a standard language model (LM) where words are considered words etc. a slightly lower n-gram order might be appropriate, but as the current data is based on characters rather than words the n-gram order is increased to capture longer words, and their spellings. Work on this has been done by Nakov and Tiedmann (2012), and their results pointed towards an order of 10 for character-based SMT.

Moses SMT was initialized with a language model of order 10, with standard smoothing (Good-Turing Discounting) and (Linear) interpolation created with SRILM then trained and tuned with the Minimum Error Rate Training (MERT) algorithm.

3 Results

The evaluation of Levenshtein and SMT is done with a quantitative analysis (Table 1) and two qualitative analyses (Table 2, Table 3).

The quantitative analysis of the normalization is illustrated in Table 1, where the total number of tokens and types normalized in the text is shown. The normalizations in Table 1 shows all normalizations that were made, regardless of their correctness.

Method	Data	Total	Norm.	%
LEV	Types	162841	54505	33,4
	Tokens	5941139	535956	9,2
SMT	Types	178292	40856	22,9
	Tokens	5939067	380577	6,4

Table 1: Data normalization percentages for Levenshtein (LEV) and SMT. Norm = Normalized.

It can be seen that a significant yet not particularly large amount of tokens were normalized, 6,4% (SMT) and 9,2% (Levenshtein), however the normalization rate for types is much higher for both methods, 22,9% (SMT) and 22,9% (Levenshtein).

The features examined in Table 2 is the following: *hv (hvad, hvilken)*, *qv (qvinna/kvinna, qvantfysik/kvantfysik)*, *dt (fasdt/fast, måladt/målat)*, *fv (hufvud/huvud, öfver/över)*, archaic preteritum conjucation of verbs ending with *-o* e.g. *gingo/gick, fingo/fick, voro/var*, these features were chosen based on their frequency and recognizability.

These words are only a subset of the totality of words with archaic Swedish spelling, changes such as $e \rightarrow ä$ and less frequent archaic spelling conventions have been ignored.

Method	Correct	Incorrect	Accuracy
LEV	895	105	89,5%
SMT	818	182	81,8%

Table 2: Normalizations of Swedish words with archaic spelling. Correct = Archaic spelling normalized correctly, Incorrect = Archaic spelling normalized incorrectly, Accuracy = $\frac{\text{Correct}}{\text{Correct+Incorrect}}$.

In Table 2, 1000 words with archaic spelling was chosen randomly, the evaluations is then done by manually checking if the words with archaic spelling is normalized to the correct modern version of the words.

True positives (TP), false positives (FP) and false negatives (FN) were identified manually. Among the 1000 candidates surnames such as *Lindqvist* are normalized to *Lindkvist*, which is correct, but in modern Swedish the archaic spelling is still being used, so these cases have been removed from the evaluation in Table 2. Many cases of French and English words appear and have been marked as FP (False Positive), also compound words such as *sandtorrt* have been marked as FP, the archaic Swedish spelling *dt* occurs in *sandtorrt*, but it is formed as the the two words *sand* and *torrt* are concatenated, it is not an actual instance of archaic spelling.

At this stage the most interesting percentage is the relationship between correct and incorrect normalizations, i.e. $\frac{\text{Correct}}{\text{Correct+Incorrect}}$, this shows how well these methods perform on words with archaic spelling, which is 89,5% for Levenshtein and 81,8% for SMT.

These numbers show how the methods perform when only relevant data is selected by the methods, however the score for the successful selection rate among the selected elements (precision) is rather low for both methods, 8,2% for Levenshtein and 7,3% for SMT. This means that out of all the words selected by the methods, 8,2% and 7,3% have archaic spelling. In contrast to this, the selection rate for the words with archaic spelling (recall) is 95.3% for Levenshtein and 92.1% for SMT, which means that out of all the words with archaic spelling, 95,3% and 92,1% of them are caught. Table 3 shows the precision and recall on types in the dataset for both methods.

Method	Precision	Recall
LEV	8,2%	95,3%
SMT	7,3%	92,1%

Table 3: Precision $\frac{\text{TP}}{\text{TP+FP}}$ and recall $\frac{\text{TP}}{\text{TP+FN}}$ for Levenshtein and SMT based on the normalization of types.

It can be observed that both methods generally output similar result for correct and incorrect normalizations on archaic words, the same trend is seen in both precision and recall where the difference is not huge, 3.2% for recall and 0.9% for precision. An extensive and complete analysis of the errors in word selection has not been done at this point, but the observed phenomenas will be addressed below.

4 Discussion

From the normalization data it has been observed that a large portion of the incorrectly normalized words is compound words, e.g. *qvinnotårar* → *kvinnoårar* (SMT), *nordljuset* → *mordlust* (Levenshtein). As shown in the first example (SMT), the first part of the compound is normalized correctly, however the second compound part is incorrectly normalized with the removal of *t*, which makes this into an incorrect normalization.

The second example (Levenshtein) shows that the entire compound word may change form, both the first and the second compound.

Levenshtein does word to word mappings and normalizes compound words as two separate words while searching for candidates. SMT on the other hand regards each word as a collection of characters and does not differentiate between compounds words and normal words.

SMT removed one letter in the first example from the compound while Levenshtein regarded its compound as two separate entities which resulted in that the two words received normalizations separately which changed the meaning of both parts instead of just one. For the Levenshtein-based approach there should be some tuning around normalizing compound words. Compound splitting can be a good thing but there could be some restrictions made to improve it.

The main issue for SMT is the different genres in the training data and the texts of August Strindberg. Common words with archaic Swedish spelling is caught, but also a few words that were not archaic but had archaic spelling features by accident were normalized, such as *sandtorrt* → *stort* (SMT). This phenomena is quite hard to solve during the normalization process, the best method would be a lexicon which consists of an updated vocabulary with a larger scope than the current one, this enables us to identify that *sandtorrt* is a compound word consisting of two modern Swedish words and not an instance of archaic Swedish spelling.

As noted many instances of *code-switching* appear in the text of Strindberg, and not all foreign words are caught in the preprocessing as these often appear randomly in the text. This lowers the representability of the test on total amount of types and tokens normalized by the methods, the accuracy is unaffected as the foreign words can be ruled out when manually checking the results.

The two problems, *code-switching* and *compound words* in conjunction is responsible for a large portion of the normalised words in both method, which as mentioned above distorts the actual number of relevant words that are normalized. Another issue is that the 5 features from Table 2 which were analyzed is not all of the words that were normalized correctly, or have archaic Swedish spelling, which means that another portion of the normalized texts is correct. Many features of archaic Swedish has been overlooked at the moment due to time limitations, as the data has to be validated by hand.

What remains to do is to Part-Of-Speech tag the text and evaluate the results of both the Levenshtein and SMT normalized versions, as well as the original texts. The suspicion is that the normalized versions will perform better than the original, but the question remains, if and how much the text normalization has done to improve the results.

Another use for POS-tagging is that the results of the POS-tagging may be an indicator of the overall performance for both SMT and Levenshtein. This can be seen from the fact that the meaning of an archaic spelled word should not be different from the modern day spelling of the word. And thus, if the POS-tag has changed it can be assumed that the word has been normalized incorrectly.[8] Some changes in the POS-tagging is wanted, for words with archaic spelling, but all other changes in the POS-tags should be regarded as incorrect, thus in performing POS-tagging all the false positive (FP) normalizations can be identified.

5 Conclusions

The accuracy of both the methods showed success, however both methods in conjunction with the training data results in a very low precision rate. Some solutions to this has been suggested, such as foreign word recognition (FWR), furthermore another data set for training that is more genre specific and closer to 19th century Swedish as well as specific methods to handle compound words better, for both Levenshtein and SMT.

In conclusion, the current methods work rather well when the input is Swedish words with archaic spelling, but for texts which contains words with modern Swedish spelling, a more complex system with additional components will be needed. This is seen clearly when comparing precision and recall of the two methods. Both methods pick up most of the instances of archaic spelling, but they also pick up a large amount of irrelevant data.

References

A. Baron and P. Rayson. 2008. *Vard2: A tool for dealing with spelling variation in historical corpora*, In Postgraduate Conference in Corpus Linguistics, Aston University, Birmingham, UK.

E. Pettersson, B. Megyesi and J. Nivre. 2012 *Rule-Based Normalisation of Historical Text a Diachronic Study*, Proceedings of KONVENS 2012 (LThist 2012 workshop), Vienna, September 21, 2012

E. Pettersson, B. Megyesi and J. Tiedemann. 2013 *An SMT Approach to Automatic Annotation of Historical Text*, Proceedings of the workshop on computational historical linguistics at NODALIDA 2013. NEALT Proceedings Series 18 / Linkping Electronic Conference Proceedings 87: 5469.

E. Pettersson, B. Megyesi and J. Nivre. 2013 *Normalisation of Historical Text Using Context-Sensitive Weighted Levenshtein Distance and Compound Splitting* , Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013); Linkping Electronic Conference Proceedings 85: 163-179

E. Pettersson 2016 *Spelling Normalisation and Linguistic Analysis of Historical Text for Information Extraction*, Studia Linguistica Upsaliensia 17. 147 pp. Uppsala: Acta Universitatis Upsaliensis, Uppsala, Sweden.

E. Pettersson 2016 *User Manual for Normalisation of Noisy Input Data using HistNorm*, Department of Linguistics and Philology. Uppsala: Uppsala university, Uppsala, Sweden.

L. Borin, M. Forsberg and L. Lönngren. 2008 *Saldo 1.0 (svenskt associationslexikon version 2)*, Sprkbanken, Gothenburg: University of Gothenburg. Gothenburg, Sweden.

P. Rayson, D. Archer, and N. Smith. 2005. *VARD versus Word A comparison of the UCREL variant detector and modern spell checkers on English historical corpora*, In Proceedings from the Corpus Linguistics Conference Series online e-journal, volume 1, Birmingham, UK.

P. Nakov, J. Tiedmann 2012. *Combining word-level and character-level models for machine translation between closely-related languages.*, In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) pages 301-305. Jeju Island, Korea. Association for Computational Linguistics.

[8] This method works with the assumption that the original POS-tag is correct. A change of tags could also mean that the normalization is successful and an incorrectly tagged word now is tagged correctly.