UNIVERSITY OF HELSINKI
FACULTY OF ARTS
DEPARTMENT OF DIGITAL HUMANITIES

Master's Thesis

Automatic Normalization of Finnish Social Media Text

Varpu Vehomäki

Master's Programme in Linguistic Diversity and Digital Humanities

Supervisor: Yves Scherrer 09.05.2022

Abstract

Faculty: Faculty of Arts

Degree programme: Master's programme in Linguistic Diversity and Digital Humanities

Study track: Language Technology

Author: Varpu Vehomäki

Title: Automatic normalization of Finnish social media text

Level: Master's thesis

Month and year: May 2022

Number of pages: 54

Keywords: Text normalization

Supervisor or supervisors: Yves Scherrer

Where deposited: Helsinki University Library

Additional information:

Abstract:

Social media provides huge amounts of potential data for natural language processing but using this data may be challenging. Finnish social media text differs greatly from standard Finnish and models trained on standard data may not be able to adequately handle the differences.

Text normalization is the process of processing non-standard language into its standardized form. It provides a way to both process non-standard data with standard natural language processing tools and to get more data for training new tools for different tasks.

In this thesis I experiment with bidirectional recurrent neural network models and models based on the ByT5 foundation model, as well as the Murre normalizer to see if existing tools are suitable for normalizing Finnish social media text. I manually normalize a small set of data from the Ylilauta and Suomi24 corpora to use as a test set. For training the models I use the Samples of Spoken Finnish corpus and Wikipedia data with added synthetic noise.

The results of this thesis show that there are no existing tools suitable for normalizing Finnish written on social media. There is a lack of suitable data for training models for this task. The ByT5-based models perform better than the BRNN models.

Contents

Acknowledgements

Many thanks to my supervisor Yves Scherrer for his support and guidance. His comments were incredibly helpful and constructive and his advice made the process of writing this thesis much easier.

I also want to thank Krister Lindén for reading my thesis, my opponents Ilmari Kylläinen and Laura Aarnio for their constructive feedback and good ideas. Lastly, I want to thank my parents for their endless encouragement.

Chapter 1

Introduction

With the rise of social media, there is plenty of text data available online, but much of it does not conform to standard language. *Text normalization* is a text processing task where non-standard language is automatically normalized to a standardised form of a language.

Normalizing non-standard text can have different benefits. Many NLP tools have not been developed for non-standard data, so normalizing the data could help these tools perform better on this data. Non-standard data as training material can also affect the performance of a system. For text-to-speech synthesis systems, text normalization is important as non-standard words can cause the synthesis to fail. Cardinal numbers and dates, for example, need to be normalized into text before the text-to-speech system is able to verbalize them.

In this thesis I experiment with bi-directional neural network (BRNN) and ByT5 models to see how the models manage to normalize Finnish social media text. The models are trained on data from the Samples of Spoken Finnish corpus (Institute for the Languages of Finland, 2014) and Wikipedia and the models are tested on manually normalized data from the Ylilauta (Ylilauta, 2015) and Suomi24 (Aller Media Ltd., 2020) corpora.

1.1 Motivation

As Partanen et al. (2019) state, it is common for Finnish speakers to use dialect when writing texts online. This can make it harder to use Finnish written online as data for NLP tasks.

The results of the MultiLexNorm shared task (van der Goot et al., 2021) show that taggers and parsers trained on social media data perform somewhat worse than those trained on canonical data. However, normalizing social media text can have a positive effect on the results of POS tagging and parsing.

Normalizing non-standard text manually is expensive and takes time, which means that the annotated data will not keep up with the rapidly changing language, especially online. The need for manually annotated or normalized data could be avoided by up-training models on data

produced by other existing models, but this has to be done specifically for every task (van der Goot, 2019b).

There are already many approaches to text normalization for historical (Bollmann, 2019), dialectal (Hämäläinen et al., 2020) and social media (Baldwin and Li, 2015) texts in many languages, but not much such research has yet been done on social media text normalization in Finnish. Partanen et al. (2019) have studied normalizing Finnish dialectal text and found that their best normalization model was able to significantly lower the word error rate of the corpus that was used, so good methods for normalizing Finnish dialects already exist.

Since normalizing online text is somewhat different from normalizing dialects, it is useful to see which approaches work best for this task. The only publicly available tool for normalizing Finnish at the moment is the Murre tool[1] by Partanen et al. (2019). Since the tool has been developed for the normalization of dialectal Finnish, it might not be the best possible tool for normalizing Finnish social media messages.

1.2 Research questions

My goal is to find out which method of text normalization works best for written Finnish online language. My main research question is "Do existing state-of-the-art normalization models work for normalizing Finnish written online?".

In addition to finding whether certain approaches work for normalizing Finnish written online, I look at the problems each approach has. I try to see if there are certain things that the models struggle with more than others.

1.3 Structure of the thesis

In chapter 2, I will describe the background of this paper and tell more about earlier work on the topic. In chapter 3, I will describe in detail the data and methods used in this thesis. Chapter 4 describes the experiments done using the data and methods described in chapter 3. Finally, in chapter 5, I will describe the results of the experiments, discuss them, and describe the conclusions I arrive to.

[1] https://github.com/mikahama/murre

Chapter 2

Background

Text normalization can be divided into subcategories by looking at the type of data that is being normalized. Veliz et al. (2021) determine two categories of text normalization, which are using normalization as a pre-processing step in connection to text-to-speech processing and normalizing user generated content (UGC). In addition to these two categories, normalization of historical texts or *modernization* can also be considered a subcategory of text normalization. Normalization of dialectal transcriptions can also be seen as a category of its own. All of these categories of normalization have their own special qualities, but also similarities between them can be found. Some methods used for normalization can work for all of these types of normalization, but some are generally used for only a certain type of normalization.

There are also different definitions for text normalization. For example, *lexical normalization* can be defined in the following way: *"Lexical normalization is the task of transforming an utterance into its standard form, word by word, including both one-to-many (1-n) and many-to-one (n-1) replacements"* (van der Goot et al., 2021). Lexical normalization generally processes text on word- or character-level, not, for example, on sentence level. While most research on normalization is focused on word-level normalization, there is also research on phrase-based normalization, like Aw et al. (2006).

Baldwin and Li (2015) show that certain kinds of normalization can be more beneficial for certain downstream tasks than others. For example, normalizing non-standard words to their standardized versions and capitalization correction is more useful for named entity recognition than for speech synthesis, which benefits more from removing unknown tokens.

2.1 Modernization

Historical text is not a clear category and historical text corpora can contain texts from the 16th century to the 2000s (Bollmann, 2018). Bollmann (2018) also states that generally the older the text is, the more difficult it is to process it.

A special challenge with modernization is spelling variation. This means that one word could historically have had many different spellings. Syntactical, morphological, and lexical changes in a language do not make it harder to develop tools to process historical language, but spelling variation does. The amount of training data needed to cover all possible spelling variants of, for example, a word with 10 spelling variants is much larger than covering a word with just one spelling (Bollmann, 2018).

Bollmann (2019) divide historical text normalization approaches into five categories. Some of these categories are also widely used in other types of text normalization. The categories are as follows:

- Substitution lists
- Rule-based methods
- Distance-based methods
- Statistical methods
- Neural models

Substitution lists are a simple approach where each word is mapped to a modern normalization. Substitution lists can not handle words or variants that are not included in the list, so on its own it is not a very effective method of normalization, but as a part of a larger system like the Norma tool it can perform very well (Bollmann, 2012).

Rule-based methods use replacement rules for normalization. These methods require a human to write these rules manually, which can be time-consuming. Rule-based systems are some of the oldest approaches to normalization and have been used for example for Old Icelandic (Fix, 1980). Hand-crafted rules have been used in combination with other approaches also for UGC, like in Barik et al. (2019). Their research shows that hand-crafted rules can improve the performance of a word embedding-based model. Rule-based methods are used in this research to generate the artificial noise in Wikipedia data described in section 4.3.2.

Distance-based methods can be very close to rule-based methods. These methods use some kind of a distance measure to find the best normalization. The goal for these methods is to get to a normalization that is the shortest possible distance away from the original word. For example, Kestemont et al. (2010) and Jurish (2010) have used distance-based methods to compare historical word forms to modern variants.

Statistical methods use probabilities to get to the correct normalization. These methods can be character-based where input is seen as sequences of characters instead of individual words. Character-based statistical machine translation has been used, for example, for the modernization of Spanish (Sánchez-Martínez et al., 2013), and Slovene (Scherrer and Erjavec, 2013).

Neural methods make use of neural nets (Bollmann, 2019). The statistical and neural normalization methods are often similar to statistical (SMT) and neural machine translation (NMT) methods (Partanen et al., 2019). Nowadays it is common for the state-of-the-art NLP tools to utilize *foundation models*, i.e., models that are trained on general data and that can be fine-tuned for a large number of downstream tasks (Bommasani et al., 2021). Some foundation models that many state-of-the-art NLP tools are based on are BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2019).

Bollmann (2019) state that out of the four models for historical normalization they compared, the Norma tool (Bollmann, 2012) performs the best on small datasets while the cSMTiser (Ljubešić et al., 2016; Scherrer and Ljubešić, 2016) performs best in other cases. Norma mixes substitution lists with a distance-based algorithm and a rule-based normalizer while the cSMTiser is a character-based statistical machine translation model.

2.2 Spoken language

Automatic speech recognition (ASR), natural language understanding (NLU) and text-to-speech synthesis (TTS) all benefit from text normalization (Mansfield et al., 2019).

Text-to-speech synthesis attempts to convert written text into speech (Macchi, 1998). Normalization is an important part of TTS systems, as these systems tend to skip non-standard-words. This causes the produced speech to be inaccurate (Sigurðardóttir et al., 2021). TTS systems can also struggle with numbers or abbreviations if they are not normalized (Ebden and Sproat, 2015). Overall, TTS systems struggle with expressions that are verbalized differently from the way they are written (Tyagi et al., 2021).

Unlike in modernization or the normalization of UGC, normalizing, for example, numbers into text is very important. However, there can be a lot of ambiguity, as Mansfield et al. (2019) point out. For example, fractions and dates can be written in the same way and which one is in question depends on the context. Verbalizing a date as a fraction is not acceptable, so they need to be normalized correctly. NLU systems, on the other hand, might normalize spoken forms into their written equivalents, so that "five" becomes "5" (Mansfield et al., 2019).

Rule-based approaches described in section 2.1 have commonly been used also for TTS systems but as said, constructing the rules is very time-consuming (Mansfield et al., 2019).

Text normalization has also been used to develop language models for the training of ASR systems (Nikulásdóttir et al., 2018). In the case of ASR, text normalization is often done to make existing data better suited for ASR. For example, numbers and abbreviations need to be spelled out. Unlike with TTS, text normalization for ASR systems is mostly done just to train models and it is not needed in the actual systems themselves.

2.3 User-generated content

Text written on social media can be referred to as user generated content (UGC). User-generated content also includes other forms of content, like audio, images etc. In this thesis I concentrate on user-generated text, specifically message board posts, while there are also other genres of UGC text, like text messages and tweets (De Clercq et al., 2013).

In their paper, van der Goot et al. (2018) describe how user-generated text differs from standard language. The taxonomy they propose shows that UGC contains anomalies that do not usually appear in standard language. Van der Goot et al. (2018) divide the anomalies in unintentional, intentional, and unknown anomalies. The category "unintentional" contains unintentional errors like typographical, spelling, splitting and merging errors. The intentional anomalies are anomalies the user has generated intentionally, like phrasal abbreviations, repetitions, shortenings of words etc.

In addition to the anomalies in van der Goot et al. (2018), Finnish online text also contains dialectal forms of words. For example, *minä* (I) often becomes *mä* and passive is used instead of first person plural, e.g., *me tulemme* (we are coming) becomes *me tullaan*.

Baldwin et al. (2013) show that different sources of UGC vary a lot from each other. They present five datasets from different sources. These datasets are: Twitter-1/2, Comments, Forums, Blogs and Wikipedia. They found that sentences in the Comments dataset are more likely to be grammatical than sentences in Forums or Blogs. They also found that Twitter data is slightly less grammatical than Forums data. The data used in my experiments would most likely be counted as Forums data as both Ylilauta[2] and Suomi24[3] are online forums. Because the lexical analysis in Baldwin et al. (2013) was only conducted for English data, the results can not be directly applied to Finnish data.

In addition to the sources mentioned in Baldwin et al. (2013), other sources for UGC are emails and text messages (Eisenstein, 2013). Email and text message data is not as easily available as, for instance, Twitter data, because emails and text messages are not publicly available (Eisenstein, 2013). According to Munro and Manning (2012) Twitter data is used significantly more often than email or text message data. There is no comparison between the use of Twitter data and the rest of the UGC source types, but Twitter is very often used (van der Goot et al., 2021), (van der Goot, 2019a).

There have been advances in the normalization of UGC for many languages as shown by the results of the MultiLexNorm shared task (van der Goot et al., 2021). The goal of the shared task was for the teams to develop normalization tools for the following languages: Danish, German, English, Spanish, Croatian, Italian, Dutch, Slovenian, Serbian and Turkish. Bilingual datasets for Indonesian-English and Turkish-German were also involved in the task. The best models

[2]https://ylilauta.org/

[3]https://www.suomi24.fi/

achieved error reduction rates of over 50 % on average.

2.4 Normalization of dialectal transcriptions

Normalization of dialectal transcriptions is not a big category of normalization, but it makes sense to mention it on its own as it differs from the other categories. This type of normalization could be seen as a mix of speech normalization and UGC normalization. Transcribed speech differs from user-generated text but both are different from standard language.

Partanen et al. (2019) show that dialectal Finnish text is quite different from standard Finnish. This happens in other languages as well, Scherrer and Ljubešić (2016) also find that only about a fifth of transcribed dialectal words in Swiss-German match their standardized forms. In section 3.1 I describe in more detail the Samples of Spoken Finnish corpus, which consists of transcribed dialectal Finnish, and the Ylilauta and Suomi24 corpora, which consist of written social media text, and the differences between these corpora.

Normalization of dialectal transcriptions has also been studied by Hämäläinen et al. (2020) who trained a bidirectional recurrent neural network (BRNN) model to normalize Finland Swedish dialectal transcriptions. Their approach is quite similar to Partanen et al. (2019), but the results are quite different. They gain a WER 28.58 at best while the best model by Partanen et al. (2019) reaches a WER of 5.73. Hämäläinen et al. (2020) state that the amount of data available should be considered when deciding the amount of context the model can handle.

Chapter 3

Data and methods

In this chapter I will present the data and methods used in this thesis.

3.1 Data

I mainly use data from the Suomi24 sentences (Aller Media Ltd., 2020) and the Ylilauta (Ylilauta, 2015) corpora. These corpora consist of messages from the online discussion boards Suomi24 and Ylilauta. I also used the Samples of Spoken Finnish (SSF) corpus (Institute for the Languages of Finland, 2014) and Wikipedia data in some of the experiments. The SSF corpus has already been normalized manually so it is a good starting point for my experiments and Wikipedia provides big amounts of text written in standard Finnish.

3.1.1 The Ylilauta Corpus

Ylilauta is an anonymous Finnish imageboard that does not require the users to sign in (Lauta Media Ltd, 2022a). The website was published in 2011 (Lauta Media Ltd, 2022b) and has around 1.5 million active users per month (Lauta Media Ltd, 2022c). The board is used for all manner of discussions and allows the users to also share pictures with each other.

Figures 3.1 and 3.2 show how discussion threads and discussions are displayed on Ylilauta.

The Ylilauta corpus contains messages from the years 2012-2014. The corpus is available as a vrt-file (Ylilauta, 2016). The data in the file is split into messages marked by the `<paragraph id=xxx>` and `<\paragraph>` tags. The sentences are marked by the `<sentence id=xxx>` and `<\sentence>` tags. The xxx here denotes the sentence and paragraph ids that were unique for every message and sentence. The data inside the sentence tags is in Conll-U format with the original sentence being in the first column. I only use this first column and discarded the rest of the data. In addition to the messages themselves, the data contains information like the time the comment was posted and the section of the forum the comment was posted to.

Figure 3.1: Screenshot of a list of discussions on Ylilauta. The navigation bar on the left has options to go to your profile, browse discussions and to find information about the page. On the right there is a list of discussions. The first word *Satunnainen* refers to the theme the discussion was posted under, in this case all the discussions are under "Miscellaneous". The text after the theme name is the title of the discussion.

Figure 3.2: Screenshot of a discussion on Ylilauta. The text on upper left, *Vinkatkaa hyviä vähemmän tunnettuja sotaleffoja sunnuntai...* (Suggest good less well-known war movies for Sunday) is the title of the discussion and the picture and text right under it are the first message. The rest of the messages are shown underneath the first one on red background.

```
<text title="TunnistusTunnustuslanka" datefrom="20141216" dateto="20141216" date="16.12.2014" clock="21:40:45" id="32825277" sec="ihmissuhteet">
<paragraph id="1">
<sentence id="1">
&gt;&gt;31924880    1   &gt;&gt;31924880    N   SUBCAT_Prop|OTHER_UNK   2   name    _
Laitetaas   2   Laitetaas   N   SUBCAT_Prop|NUM_Sg|CASE_Nom|CASECHANGE_Up|OTHER_UNK 4   nn  _
taas    3   taas    Adv _   2   advmod  _
uutta   4   uusi    A   NUM_Sg|CASE_Par|CMP_Pos 0   ROOT    _
pystyyn 5   pystyyn Adv _   4   advmod  _
kun 6   kun C   SUBCAT_CS   8   mark    _
ei  7   ei  V   SUBCAT_Neg|PRS_Sg3|VOICE_Act    8   neg _
näkynyt 8   näkyä   V   NUM_Sg|CASE_Nom|VOICE_Act|PCP_PrfPrc|CMP_Pos    4   advcl   _
olevan  9   olla    V   NUM_Sg|CASE_Gen|VOICE_Act|PCP_PrsPrc|CMP_Pos    8   xcomp   _
vielä   10  vielä   Adv _   9   advmod  _
.   11  .   Punct   _   4   punct   _
</sentence>
</paragraph>
</text>
```

Figure 3.3: Screenshot of how the Ylilauta data is structured in the vrt-files.

The split into sentences in the Ylilauta corpus is not perfect. The splitting was apparently done by using period, question mark and exclamation mark as sentence boundary markers within messages. This has not provided perfect results, as sometimes the sentence boundary is not marked by ".". In these cases, two or more sentences have been marked as one sentence. There are also cases where one sentence was split into two or more due to erroneous use of ".".

3.1.2 The Suomi24 Sentences Corpus

Suomi24 is a Finnish message board. Like Ylilauta, Suomi24 does not require users to sign up. The users can post messages to existing discussions or start new ones. The discussions are categorized into themes that are categorized into subcategories (Lagus et al., 2016).

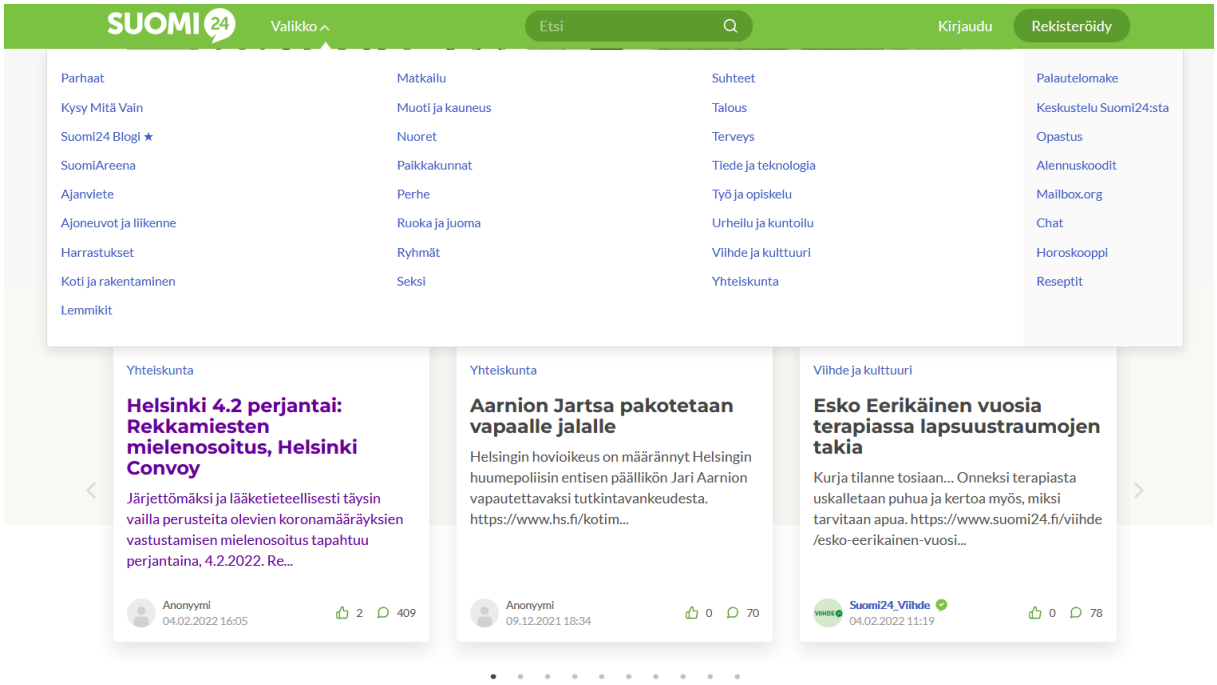

Figure 3.4: Screenshot of the current interface of Suomi24. On the upper half is the navigation bar that shows the different themes available. On the bottom there are three recommended discussions.

The Suomi24 Sentences Corpus contains all forums on the Suomi24 website from 2001. The corpus is updated twice every year and at the time of writing there was data from 2001 to 2020. I use a version of the corpus containing data from 2001 to 2017. This is because some of the newer data is still in beta state and might contain parsing errors etc. The corpus is divided into individual files, each file containing messages posted in a certain year (Aller Media Ltd., 2020)

The format of the data in the corpus is very similar to the Ylilauta Corpus. Figure 3.4 shows the main page of the Suomi24 forum and figure 3.5 shows how discussions are displayed on the forum. Each comment has its own comment ID and is marked by a `<text =...>` and a `</text>` tag. The `<text>`-tag also contains information like the time the comment was posted, the username of the author, the name of the thread the comment is in and so on. Each sentence in each comment is marked with the `<sentence id=xxx>` and `<\sentence>` tags.

Just like the Ylilauta Corpus, the Suomi24 Sentences Corpus also contains messages in English, URLs and other noise. There are also sentence splitting errors that cause several sentences to be marked as one.

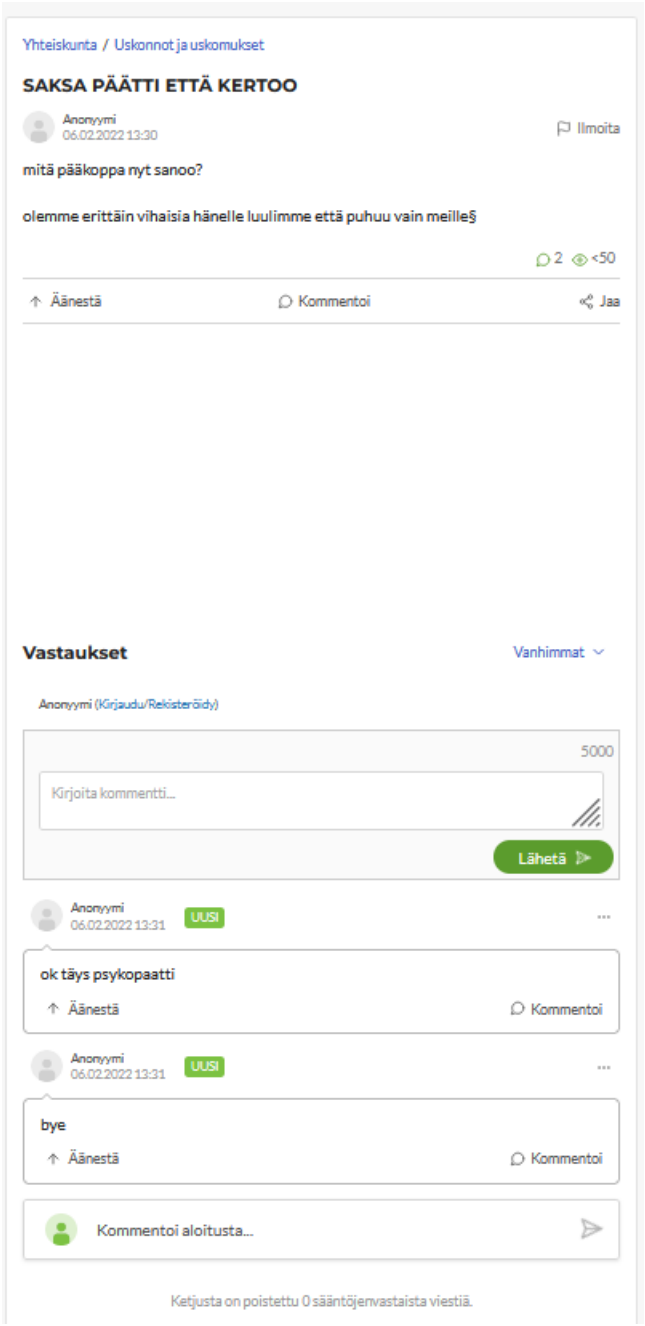

Figure 3.5: Screenshot of how discussions are displayed on Suomi24. The discussion is under the theme *Yhteiskunta* (Society) under the subcategory *Uskonnot ja uskomukset* (Religions and beliefs). The text in bold high up in the picture is the title of the discussion, right under it is the user who started the conversation, the time the conversation was started and the starter's message. Underneath the title *Vastaukset* (Replies) is the message box where the user can write their own reply to the first message. Underneath the reply bar are the rest of the replies to the thread.

3.1.3 The Samples of Spoken Finnish Corpus

The Samples of Spoken Finnish corpus (Institute for the Languages of Finland, 2014) is based on a series of booklets published by the Institute for the Languages of Finland in 1978–2000.

```
<text comment_id="0" date="2001-01-01" datetime="2001-01-01 01:30:00" author="Honda" parent_comment_id="0" quoted_comment_id="0" author_logged_in="n"
nick_type="anonymous" thread_id="19455" time="01:30:00" title="Hyvää uutta vuotta kaikille Hondailijoille" topic_nums="3258,1109,6254,2" msg_type="thread_start"
topic_name_leaf="Honda" topic_name_top="Ajoneuvot ja liikenne" topic_names="Ajoneuvot ja liikenne &gt; Autot &gt; Automerkit &gt; Honda"
topic_names_set="|Ajoneuvot ja liikenne|Automerkit|Autot|Honda|" topic_nums_set="|1109|2|3258|6254|" topic_adultonly="n" datefrom="20010101" dateto="20010101"
timefrom="013000" timeto="013000" id="19455:0" author_v1="Honda" author_name_type="user_nickname" author_nick_registered="n" author_signed_status="0"
thread_start_datetime="2001-01-01 01:30:00" filename_vrt="s24_2001_01.vrt" parent_datetime="" datetime_approximated="n" empty="n" filename_orig="threads2003a.vrt"
origfile_textnum="17841">
<paragraph id="1" type="title">
<sentence id="1">
Hyvää   1   hyvä    hyvä    A   NUM_Sg|CASE_Par|CMP_Pos|CASECHANGE_Up   3   amod    _   1   |hyvä..jj.1|
uutta   2   uusi    uusi    A   NUM_Sg|CASE_Par|CMP_Pos 3   amod    _   2   |uusi..jj.1|
vuotta  3   vuosi   vuosi   N   NUM_Sg|CASE_Par 0   ROOT    _   3   |vuosi..nn.1|
kaikille    4   kaikki  kaikki  Pron    SUBCAT_Indef|NUM_Pl|CASE_All    5   det _   4   |kaikki..pn.1|
Hondailijoille  5   Hondailijoille  Hondailijoille  N   SUBCAT_Prop|NUM_Pl|CASE_All|CASECHANGE_Up|OTHER_UNK 3   nommod  SpacesAfter=\n\n    5   |Hondailijoille..nn.1|
</sentence>
</paragraph>
```

Figure 3.6: Screenshot of how the Suomi24 data data is structured in the vrt-files.

These booklets consisted of transcribed dialectal speech. The corpus contains annotated speech from different dialectal areas of Finland.

All of the sentences have been normalized according to the guidelines made for this task (Vilkuna, 2014). The normalizations are aligned with the unnormalized text on token-level. The corpus also has both a more detailed transcription of the speech and a simplified version. For my experiments I use the simplified transcriptions, as they are more similar to Finnish written online. Table 3.1 shows an example of the SSF data.

	Source	**Normalization**
1	täyty	täytyi
2	,	,
3	mu-	-
4	määki	minäkin
5	munta	monta
6	kerta	kertaa
7	oli	olin
8	hakkaamassa	hakkaamassa
9	ko	kun
10	oli	oli
11	hakokuarmia	hakokuormia
12	siinä	siinä
13	nin	niin
14	,	,
15	hakkaamasa	hakkaamassa
16	niitä	niitä
17	sit	sitten
18	.	.

Table 3.1: Example of a sentence and its normalization in the SSF data.

3.1.4 Synthetic Wikipedia data

For some of the experiments I also use Wikipedia with synthetic noise in the training data. The synthetic data generation was proposed by Samuel and Straka (2021). I download a Finnish Wikipedia dump[4] and pre-process the data by tokenizing it and filtering out lines that are less than 32 or more than 160 characters long. I add the synthetic noise by using the noise-generation code by Samuel and Straka (2021). All of the code for downloading and processing the Wikipedia dump and adding the noise is based on code by Samuel and Straka (2021). I make some modifications to the code to make it work for Finnish, but the approach is largely the same. The noise is added in the following ways:

- Words in the Wikipedia data that are present in the normalized SSF training data are replaced with an unnormalized variant found in the SSF data proportionally to the number of occurrences.
- Character level modifications were made with a probability estimated from Finnish data beforehand.
- To simulate typographical errors, the code skips some characters and changes some characters to other characters that are usually close to each other on the keyboard.
- Words are split or merged with a probability estimated beforehand.
- Characters are repeated more than two times or omitted completely.

The probabilities for the changes are estimated from the Ylilauta and Suomi24 data. This was done by choosing random sentences from the datasets, then counting the occurrences of the different anomalies and dividing this number by the number of total words in the randomly chosen data. There was no need to add new rules to add the noise for now, as the rule to substitute words with unnormalized words from the training data covers most of the special cases for Finnish, such as substituting the word *hän* (3rd person singular pronoun) with the dialectal version *hää*. Table 3.3 shows the error types and their estimated likelihoods.

3.1.5 Synthetic Samples of Spoken Finnish data

I also generate new synthetic data using the normalization of the SSF corpus. The process is the same as described in section 3.1.4 but it is applied on the SSF data. This is done because the Wikipedia data is of different domain than the test sets. Online forum texts are usually some type of dialogue between people and that makes the speech data of the SSF corpus more similar to the test data.

[4] https://dumps.wikimedia.org/fiwiki/20211220/fiwiki-20211220-pages-articles-multistream.xml.bz2

	Synthetic	Original
1	1900-luvun	1900-luvun
2	alun	alun
3	Kivi-renessanssista	Kivi-renessanssista
4	alkkate	alkaen
5	hää	hän
6	on	on
7	olt	ollut
8	Suamen	Suomen
9	kansalliskirjailija	kansalliskirjailija
10	.	.

Table 3.2: Example of a sentence with added noise from the Wikipedia data.

Error type	Estimated probability of error
multiplier	1.0
typo	0.004126547
joined words	0.002751032
split words	0.004126547
missing vowels	0.001100413
remove repeated letters	0.000692042
repeated letters	0.000550206
first letter of lowercased word capitalized	0.010671256
whole lowercased word capitalized	0.00277585
whole capitalized word lowercased	0.035335689
whole uppercased word lowercased	0.068965517
uppercased word capitalized	0.031914894
replaced e	0.005780347
replaced end i	0.034120735
replaced end n	0.007960199
replaced a	0.002614379
replaced t	0.005154639
replaced i	0.000615385
replaced s	0.000875657
replaced hyphen	0.000275103

Table 3.3: Estimated error likelihoods.

3.1.6 Manual normalization

The messages in the Ylilauta and Suomi24 Sentences corpora are not normalized, so I randomly chose 200 sentences from the Ylilauta corpus and 170 sentences from the Suomi24 Sentences corpus and manually normalized them to use for testing[5]. The Ylilauta corpus was in one file,

[5]The test set is available on request from the author

while the Suomi24 Corpus was split into several files. I chose ten sentences from each of the 17 Suomi24 files to get data from different time periods. The data split is not balanced as the Suomi24 Sentences corpus is much bigger than the Ylilauta corpus. Additionally, the earlier years of the Suomi24 Sentences have less data than the more recent ones. The aim of this study is not to examine the differences between the two corpora, so balancing the amount of data from each corpus is not necessary.

Due to the cleaning of numbers in the pre-processing state, some sentences lack numbers. Some of the sentences also should be seen as separate sentences, but due to punctuation errors or errors in the parsing, they were marked as one sentence. I did not split any sentences manually but normalized these cases as if they were one sentence.

Another problem with the Ylilauta corpus is that it also involves messages in English. During the manual normalization process I skip all messages written completely in English or any other language than Finnish, but I do include sentences with some English words/expressions in them. There are also some HTML tags and URLs in the data. I delete the HTML tags when I extracted the sentences from the data, but I left the URLs.

The normalization proved to be more challenging than what I had initially thought. I followed the normalization guidelines (Vilkuna, 2014) used in normalizing the Samples of Spoken Finnish corpus (Institute for the Languages of Finland, 2014) as close as possible. These guidelines were helpful, but as they were made for normalizing dialectal speech transcriptions, they did not account for phenomena that only exist in written text. It has to be pointed out that the normalization of the Samples of Spoken Finnish corpus was done to make searching the corpus easier, so the goal of the normalization was somewhat different from mine and thus not all of the guidelines were suitable for my purposes. I also followed the annotation style from van der Goot et al. (2021) when the Samples of Spoken Finnish guidelines were not adequate.

Here are some of the guidelines I followed:

- **Interjections and punctuation:** Interjections are left untouched so *hahaha* remains *hahaha* and does not become, for example, *haha*. Punctuation is corrected when it comes to hyphens, for example, *Twinrix nimisen* (called Twinrix) becomes *Twinrix-nimisen*, but dots, exclamation marks and commas are left untouched.

- **Non-words and emoticons:** Non-words and emoticons are left untouched, XDDD -> XDDD.

- **Usernames, hashtags and URLs:** Usernames, hashtags and URLs are left untouched

- **Foreign language text:** Sentences entirely in foreign languages are left out of the normalization, individual words in other languages are left untouched. (Mostly English).

- **Agreement:** Agreement between subject and verb is corrected, *me ollaan -> me olemme* (we are). Case markers with transitive verbs are also adapted.
- **Capitalization:** Named entities are always capitalized, beginnings of sentences are in accordance with the capitalization of the training data.
- **Non-standard words:** Words that are not used in standard language are not normalized to their standard language equivalents. Only possible inflection or spelling mistakes are corrected.
- **Word order:** Word order is not corrected even if the word order in the sentence does not follow the word order used in standardized Finnish.

	Original	**Target**
1	ne	ne
2	cruiserit	**Cruiserit**
3	mitä	mitä
4	taidat	taidat
5	haluta	haluta
6	on	**ovat**
7	isoja	isoja
8	ja	ja
9	painavia	painavia
10	pyöriä	pyöriä
11	,	,
12	kuristettuna	kuristettuna
13	voit	voit
14	jonkun	jonkun
15	saada	saada
16	a2	**A2-kortille**
17	kortille	
18	sopivaksi	sopivaksi
19	.	.

Table 3.4: An example of a manually normalized sentence, edited words are marked with bold.

Due to inconsistencies with capitalization in the training datasets that are used, I have two different test sets, one with the first words of sentences capitalized and one with only named entities capitalized. The test set used in each experiment depends on which guidelines the training data follows.

It should be noted that even with these guidelines, there are still cases that are open to interpretation. It is common to have more than one annotator when doing manual normalization to ensure the reliability of the normalization. These guidelines are also not the only correct option,

and some changes could be made to make them fit the task better. For example, correcting the agreement between subject and verb could have been left untouched, as the normalization is done on word-level.

	Ylilauta	**Suomi24**
Characters	26353	20215
Words	3542	2871
Sentences	200	170
Average sentence length	18.855	18.0
Edited words	14.26%	13.33%

Table 3.5: Test set statistics

Table 3.5 shows some statistics on the test sets. There are some slight differences between the sets, such as that the Suomi24 data is a little less noisy than the Ylilauta data. Overall, the datasets are quite similar with regard to average sentence length and noisiness.

3.2 Models

3.2.1 Leave-as-is baseline

I use the leave-as-is (LAI) model as a baseline. In this baseline, no changes are made in the source text. The goal is to reach evaluation scores that are better than the ones this baseline produces.

3.2.2 Murre normalizer

Murre is a library for normalizing dialectal Finnish. It includes the chunk-level BRNN normalizer described in Partanen et al. (2019). The normalizer is trained on the Samples of Spoken Finnish corpus, and it is largely similar to the default BRNN in the OpenNMT toolkit (Klein et al., 2017). The model has two encoding and decoding layers, a general global attention model by Luong et al. (2015). Murre is a character-level model so each character is fed to the model individually, separated by white-space.

3.2.3 BRNN

I train several BRNN models. BRNNs are machine learning models based on recurrent neural networks (RNN). The advantage of BRNNs compared to regular RNNs is that they can process both the past and future information at each time frame of a sequence (Schuster and Paliwal, 1997).

The models are implemented using OpenNMT (Klein et al., 2017) and they are very similar to the Murre normalizer described in the previous section, i.e., there are two encoding and decoding layers and the attention model is the general global attention model by Luong et al. (2015). The data is also fed to the models in the same format.

Attention enables a model to focus on the features that are relevant in a sequence. The global attention model by Luong et al. (2015) attends to all of the words in a source sequence. This way, the mechanism produces a context vector to help with the prediction of the target word.

I train the BRNN models on different mixes of the SSF and the synthetic Wikipedia data, the different experiments are described in more detail in section 4.2.

3.2.4 ByT5

I also experiment with an approach based on the winning entry to the Multilingual Lexical Normalization (MultiLexNorm) (van der Goot et al., 2021) shared task by Samuel and Straka (2021). The model was chosen because the results of the shared task outperformed previous state-of-the-art models. Another reason for choosing this approach is that Samuel and Straka (2021) were able to train ByT5-based models for different datasets and they all performed well. Even though Finnish was not included in the shared task, I decided to choose an approach that has shown potential in normalizing multiple languages rather than choosing a model solely developed for normalizing one language that is not Finnish.

The model is based on the ByT5 foundation model (Xue et al., 2021b). ByT5 is a token-free model that processess sequences of bytes of UTF-8 encoding (Xue et al., 2021a). ByT5 is similar to the mT5 model by Xue et al. (2021b) it is based on. The difference between mT5 and ByT5 is that mT5 processes SentencePiece tokens and ByT5 processes UTF-8 bytes. In ByT5 the encoder is 3 times deeper than the decoder, while mT5 has equally deep encoder and decoder stacks. The main goal of developing the ByT5 foundation model was to modify a token-based model to develop a token-free model (Xue et al., 2021a). ByT5 was pre-trained on 108 languages on the mC4 corpus introduced by Xue et al. (2021b).

Samuel and Straka (2021) used ByT5 by taking the small variant of the ByT5 model and pre-training it on Wikipedia data with added synthetic noise. After the pre-training phase they fine-tuned the model on the MultiLexNorm data. The MultiLexNorm data is gathered from social media platforms in several different languages (van der Goot et al., 2021). Samuel and Straka (2021) added the pre-training step, as their experiment with just fine-tuning the model did not provide very good results. It has to be noted that generally pre-training refers the unsupervised training of a language model but here both pre-training and fine-tuning are supervised. The steps are called pre-training and fine-tuning due to the difference in the data used in each step.

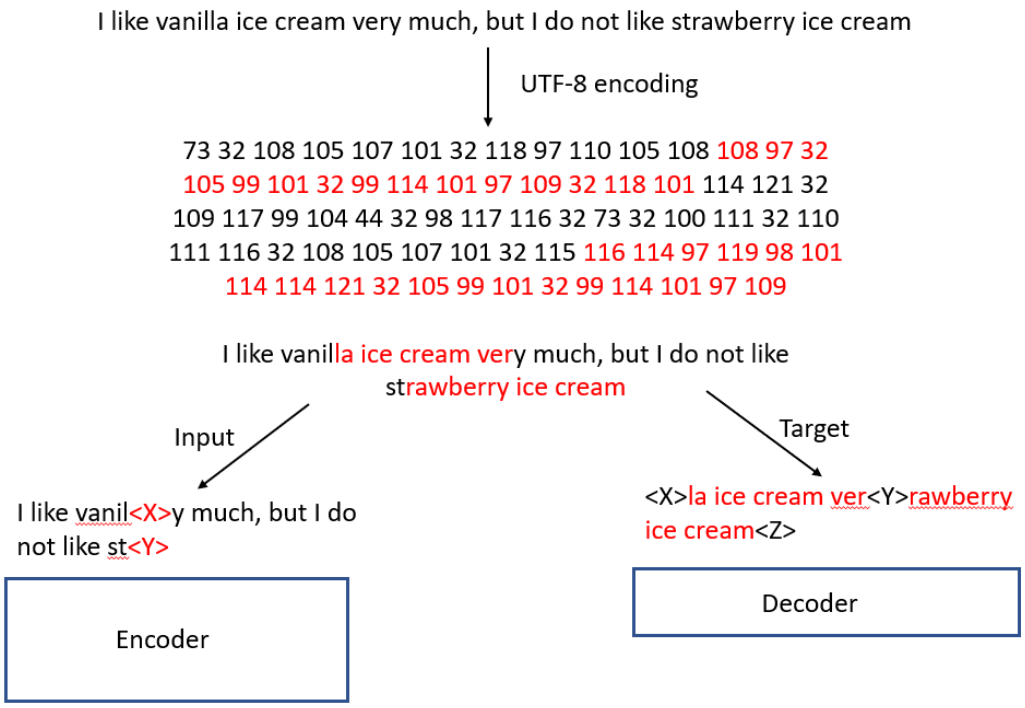

Figure 3.7: The architecture of ByT5 and an example of how the pre-training was conducted. <X>, <Y>, <Z> are sentinel tokens. The figure is based on a figure by Xue et al. (2021a).

3.3 Evaluation

To evaluate the performances of the normalization systems I use five evaluation metrics: Word Error Rate (WER) and Error Reduction Rate (ERR), accuracy, over-normalization rate (OR) and under-normalization rate (UR). I use several different metrics to make comparisons between papers easier and to get some more insight into the results.

WER is a metric that is often used to evaluate text normalization (Partanen et al., 2019). It is obtained by dividing the minimum number of deletions, substitutions and insertions needed to get to the gold standard with the number of words in the gold standard text (van der Goot, 2019b). The lower the WER the better the result.

WER is calculated in the following way:

$$WER = \frac{substitutions + deletions + insertions}{n\_words\_in\_reference}$$

The WER is calculated on word-level, which means that both a completely wrong normalization and a normalization that is just one character off are considered one substitution.

Accuracy is calculated in the following way:

$$accuracy = \frac{n\_correctly\_normalized\_words}{n\_all\_words}$$

Accuracy is also a commonly used evaluation metric, but it has its problems. Comparing accuracies of different models tested on different data is not always useful as a good accuracy on a certain dataset may not be that good on another. I use this metric mostly to compare the different methods to the LAI baseline.

I also use Error Reduction Rate (ERR), an evaluation metric that attempts to correct some of the weaknesses of some other commonly used evaluation metrics. ERR was developed by van der Goot et al. (2021) for the following reasons:

- Interpreting other metrics, like F1 score or accuracy, can be challenging because they do not actually show how much of a problem there is left to solve.
- WER can be overly complicated for lexical normalization because word order is not changed.
- Using accuracy makes it hard to make comparisons between different corpora because the amount of normalization needed varies.

ERR is calculated in the following way:

$$ERR = \frac{\%accuracy - \%words\_no\_normalization\_needed}{100 - \%words\_no\_normalization\_needed}$$

The ERR score is usually between 0.0 and 1.0 with 0.0 being the same as the LAI baseline and 1.0 being the perfect score. A negative ERR score indicates that the model makes wrong normalizations more often than correct normalizations. ERR's weakness is that is does not show whether a system normalizes too much or too little of the input. If that information is needed, precision and recall should be used in addition to ERR (van der Goot, 2019b). ERR was also used in the MultiLexNorm shared task van der Goot et al. (2021), so using the metric helped with comparing the results.

The OR is calculated in the following way:

$$OR = \frac{n\_modified\_words}{n\_words\_no\_normalization\_needed}$$

The UR is calculated in the following way:

$$UR = \frac{n\_not\_modified\_words}{n\_words\_normalization\_needed}$$

Both the OR and the UR should be as low as possible. The reason for using the OR and the UR is to get some more insight into the other metrics. Using the OR and the UR makes it possible to easily see whether a model either over-normalizes or under-normalizes. This in turn can help with making the models better.

Chapter 4

Experiments

4.1 Murre normalizer

To have a baseline for how existing systems perform on the data, I test Murre library normalizer on the test data. This requires no training as the normalizer available in the library has already been trained. The model is described in more detail in section 3.2.2.

4.2 BRNN

I train several BRNN models on different data setups. I train two BRNN models using only the synthetic Wikipedia data. The synthetic Wikipedia data is described in more detail in section 4.3.2. I also train three models using different mixes of the synthetic Wikipedia data and the SSF data.

I use the following data setups:

1. 40k sentences of the Wikipedia data as the training set, 4,000 sentences of the Wikipedia data as the validation set. This model will be referred to as BRNN Wiki.

2. The 4,000 sentences of the Wikipedia data from the validation set of setup 1 as the training set and 600 sentences of the Wikipedia data as the validation set. This model will be referred to as BRNN Wiki Small.

3. The same training set as in setup 1 + additional 40k sentences of the SSF data, 1k sentences of SSF data and 1k Wikipedia data in the validation set. This model will be referred to as BRNN Mixed.

4. The same training set as in setup 1 + additional 20k sentences of the SSF data, same validation set as in setup 3. This model will be referred to as BRNN Mixed Small.

5. 20k lines of synthetic SSF data + 40k lines of the Wikipedia data training set and 1k synthetic SSF data + 1k lines of the Wikipedia data in the validation set. This model will be referred to as BRNN Synthetic.

I use the same model settings as Partanen et al. (2019), which means that I train the models for 100,000 steps, use the general global attention model by Luong et al. (2015) and that the encoder and decoder both have 2 layers.

I also follow approach of chunking the input data into chunks of three words by Partanen et al. (2019). Because this is a character level model, characters are separated by whitespace. Sentence boundaries are marked with the <BLANK> token and word boundaries are marked by _. Table 4.1 shows what the input data looks like.

In cases where the words have been merged together in the source, the words are separated in the target, meaning that while the source is always in chunks of three (or less), there can be bigger chunks in the target. An example of this can be seen on line 5 of table 4.1, where the three words *on* (is), *espanjaa* (Spanish) and *ja* (and) have been merged into one word *onespanjaa* in the source. In the target the words are separated and thus the target contains a chunk of five words.

For BRNN Mixed I have to edit the SSF data to match the Wikipedia data better. There is a disparity between the capitalization of beginnings of sentences between the two datasets. Because the beginnings of sentences are capitalized in the Wikipedia data, I build a pre-processing script to capitalize the first words of every sentence in the SSF data. I only capitalize the actual beginnings of sentences, so if a sentence begins with a non-word, like the "-" character, the following word is not capitalized.

Because the data seems to be the biggest limitation to the models' performances, I use some different data for the BRNN Synthetic model. In addition to the synthetic Wikipedia data, I use the synthetic SSF data described in section 3.1.5.

	Source	**Target**
1	C o n c e p c i ó n _ o u m _ B i o b í o n	C o n c e p c i ó n _ o n _ B i o b í o n
2	a l u e e n _ p ä ä k a u p u n k i _ C h i l e s s ä	a l u e e n _ p ä ä k a u p u n k i _ C h i l e s s ä
3	.	.
4	<BLANK>	<BLANK>
5	N i m i _ o n e s p a n j a a j a _ t a r k o i t t a a	N i m i _ o n _ e s p a n j a a _ j a _ t a r k o i t t a a
6	N E I T S Y T _ M a r i a _ n	N e i t s y t _ M a r i a n _
7	p e r i s y n n i t ö n t ä _ s i k i ä m i s t ä _ .	p e r i s y n n i t ö n t ä _ s i k i ä m i s t ä _ .
9	<BLANK>	<BLANK>
10	K a u p u n g i n _ a s u k k a i s t a _ k ä Ö t E t t Ä Ä V	K a u p u n g i n _ a s u k k a i s t a _ k ä y t e t ä ä n
11	n i m i ä _ " _ p e n q u i s t a	n i m e ä _ " _ p e n q u i s t a
12	" _ .	" _ .
13	<BLANK>	<BLANK>

Table 4.1: Example of input data of the BRNN model

4.3 ByT5

I train three ByT5 models with the following three setups:

1. No fine-tuning stage 1 –> fine-tuning stage 2 with SSF data. This model will be referred to as ByT5 SSF.

2. Fine-tuning stage 1 on 1 million lines of synthetic Wikipedia data –> fine-tuning stage 2 on another 4,000 lines of the synthetic Wikipedia data. This model will be referred to as ByT5 Wiki.

3. Fine-tuning stage 1 on synthetic Wikipedia data –> fine-tuning stage 2 on SSF data. This model will be referred to as ByT5 Mixed.

Fine-tuning stage 1 refers to what Samuel and Straka (2021) call "pre-training" and fine-tuning stage 2 refers to what they call "fine-tuning". The training process is the same in both stages, but in fine-tuning stage 1 the synthetic noise is generated during the training while in stage 2 there is no noise-generation. Samuel and Straka (2021) call fine-tuning stage 1 "pre-training" and fine-tuning stage 2 "fine-tuning" to distinguish between training on synthetic and non-synthetic data. This division does not work for my thesis, because I use synthetic data in both stages in some of the experiments. The code all of the experiments are based on can be found on Github[6].

4.3.1 ByT5 SSF

I start my experiments with the ByT5 foundation model by training the ByT5 model on the SSF corpus (Institute for the Languages of Finland, 2014) data. I follow the approach of Samuel and Straka (2021) and construct a ByT5 input for every input word, marking the beginning and the end of a word with a sentinel token. This means that every word is normalized independently.

In my first experiment I use the Samples of Spoken Finnish data split into a training and a development set. This means that this experiment does not contain the fine-tuning stage 1, so no synthetic noise is generated in any data during the training. For the test set I use the Ylilauta and Suomi24 data I normalized myself.

I train the model for 39 epochs with total batch size of 128. I set the maximum length of the encoder at 200 and the maximum length of the decoder at 32. Samuel and Straka (2021) trained their model for 50 epochs but I stop the training early, because the validation loss no longer significantly decreases and training the model is very time-consuming.

[6]https://github.com/ufal/multilexnorm2021

4.3.2 ByT5 Wiki

For the second experiment I add the fine-tuning stage 1, which means that synthetic noise is generated in the Wikipedia data during training.

Because using all of the 5,952,553 lines of Wikipedia data in the Wikipedia dump would be so time-consuming, I only use a smaller portion of the data to train the model. I take a million lines of the data and use that for fine-tuning stage 1. As the SSF data does not seem suitable for this task, I decide to use the Wikipedia data for the fine-tuning phase too. I take 4000 lines of the Wikipedia data with added noise for the fine-tuning training set. These 4000 lines cover about 100 000 words, which is a little more than the biggest dataset in the shared task (van der Goot et al., 2021) the original ByT5 normalization models were developed for. I do not use a separate development set out of the Wikipedia data, but instead I use 10% of the training data for the development test set. I also make sure that the million lines of data used for pre-training do not overlap with the 4000 lines of the training data.

I train this model with the same parameters as the first one, also for 39 epochs.

4.3.3 ByT5 Mixed

For the third experiment I train a model using both fine-tuning steps described in section 4.3.2. This time I use the SSF data for fine-tuning stage 2.

As described in section 4.2, there are some differences between the normalizations in the Wikipedia and the SSF data. The biggest difference between the datasets is that the SSF sentences are not capitalized but the sentences in Wikipedia data are.

Chapter 5

Results & discussion

In this chapter I present the results of the different experiments. Table 5.1 shows the LAI baselines on the lowercased test sets. Because of differences in training data, test sets with capitalized first words are also used. The LAI baselines of these test sets can be seen on table 5.2. The tables show that the accuracies, WERs and URs are somewhat different between the test sets, and this should be taken into account when looking at the results of the different models.

As discussed in section 3.3, for a model to be of any use, the accuracy of the model should be better than the baseline accuracy. The ERR baseline is 0, so the ERR should be positive, a negative ERR means that the normalization is noisier than the original text. The WER should be lower than the baseline for the model to be useful.

	Ylilauta	**Suomi24**	**Both**
Accuracy:	85.77	86.67	86.17
ERR:	0	0	0
WER:	14.77	14.25	14.53
OR:	0	0	0
UR:	14.23	13.33	13.83

Table 5.1: LAI baseline on the lowercased test sets. Results based on these sets will be marked with †.

Table 5.3 shows the WERs of all of the models on both the Ylilauta and Suomi24 data combined. ByT5 Wiki is the best of all the models and all of the ByT5 models perform better than any of the BRNN models. In the rest of this chapter, I will present the results in more detail and discuss the differences between the models.

The results are presented in order of model complexity. First, I will present the results of the Murre normalizer, as it is an off-the-shelf model and did not need to be trained. Then I will present the results of the different BRNN models as the models are very similar to the Murre

	Ylilauta	Suomi24	Both
Accuracy:	89.03	90.80	89.82
ERR:	0	0	0
WER:	11.02	9.31	10.26
OR:	0	0	0
UR:	10.97	9.20	10.18

Table 5.2: LAI baseline on the uppercased test sets. Results based on these sets will be marked with *.

Model	WER
Murre:	21.10†
BRNN Wiki:	13.91*
BRNN Wiki Small:	16.14*
BRNN Mixed:	11.72*
BRNN Mixed Small:	12.08
BRNN Synthetic:	12.32*
ByT5 SSF:	14.00†
ByT5 Wiki:	**8.84***
ByT5 Mixed:	9.32*

Table 5.3: WERs of all models on both test sets.

normalizer. Lastly, I will present the ByT5 results as the approach is more complex and less similar to the other models.

5.1 Murre results

	Ylilauta	Suomi24	Both
Accuracy:	78.71	81.42	79.92
ERR:	-49.60	-39.37	-45.21
WER:	22.06	19.64	21.10
OR:	12.63	10.85	11.84
UR:	4.47	3.39	3.99

Table 5.4: Results of Murre library normalizer on Ylilauta and Suomi24 data †

The Murre normalizer does not perform very well on the test sets. Table 5.4 shows that the WER of the model is 21.10. In the original paper the BRNN model reached a WER of 5.73 on dialectal data (Partanen et al., 2019). Based on these results it is obvious that a model for social media data normalization for Finnish is needed and that existing models are not sufficient, even though their performance on the data they were developed for is good.

The Murre normalizer normalized certain characters like *f, d, g* and *b* to *v, t, k* and *p* respectively. These letters do appear in Finnish, but they are relatively rare and used more often in loan words. A couple of examples of this: *fiktiiviset* -> *viktiiviset*, *burton* -> *purton*, *götze* -> *köteen*, *costa* -> *kosta*. An example of this can also be seen in table 5.18 on line 5 where *euforinen* has been normalized as *euvorinen* although no normalization would actually have been needed.

	Original	**Prediction**	**Target**
1	Joo	joo	joo
2	,	,	,
3	just	**just**	juuri
4	totahan	**totahan**	tuotahan
5	kaikki	kaikki	kaikki
6	naiset	naiset	naiset
7	haluaakin	**haluaakin**	haluavatkin

Table 5.5: An example of a sentence normalized by the Murre normalizer. Prediction errors are in bold.

One very apparent problem the Murre normalizer had is that it does not correctly normalize URLs. The normalizer has not been trained on data that contains URLs so to make the evaluation fairer for the model I used a post-processing script that copied URLs from the source text into the prediction. There are only six URLs in the test sets and all of them are in the Ylilauta data.

The OR and UR show that overall it is more common for the model to over-normalize. Murre has an OR of 11.84 which is significantly higher than the UR of 3.99. There are also cases where the sentences are not normalized enough. Table 5.5 shows an example of a sentence that has not been normalized at all by the model.

5.2 BRNN results

In this section I will present the results by the BRNN models. For the evaluation I have chosen the checkpoints with the best validation accuracy reported during training.

5.2.1 BRNN Wiki

The BRNN Wiki model performs noticeably better than Murre, but the results are still not good enough for the normalizations to be useful. The model seems to improve the normalizations of especially the Ylilauta data compared to the Murre normalizer, but the ERR shows that the normalizations are still far from being useful.

	Ylilauta	Suomi24	Both
Accuracy:	85.88	87.79	86.73
ERR:	-28.72	-32.70	-30.32
WER:	14.34	13.37	13.91
OR:	5.57	5.14	5.38
UR:	6.72	3.74	5.40

Table 5.6: Results of BRNN Wiki *

This model handles foreign words better than Murre and it does not have the same problems with the letters *f, d, g* etc.

5.2.2 BRNN Wiki Small

	Ylilauta	Suomi24	Both
Accuracy:	83.77	84.32	84.02
ERR:	-47.95	-70.34	-56.97
WER:	16.31	15.92	16.14
OR:	6.72	7.94	7.27
UR:	7.09	4.06	5.74

Table 5.7: Results of BRNN Wiki Small *

The second BRNN model performs better than the Murre normalizer but worse than BRNN Wiki. This implies that larger amounts of data are beneficial for the performance of a model.

The model struggles with punctuation, especially with exclamation marks. Table 5.15 shows how the model struggles with repeated exclamation marks and normalizes only one exclamation mark out of 24 correctly.

5.2.3 BRNN Mixed

The first experiment with Wikipedia data mixed with SSF data gave better results than the other BRNN setups. Table 5.8 shows that especially the ERR gets considerably better, but it still does not reach any positive values. This means that the model is still not good enough to be useful.

According to table 5.8 the model tends to under-normalize rather than over-normalize. Table 5.9 shows that the model has learned to correct the dialectal *sulle* (to you) into the standard Finnish *sinulle*.

As stated in section 4.2, the data was processed to be more homogeneous when it comes to capitalization. Despite this, the model did not learn to consistently capitalize every first word of a sentence. The model follows the capitalization of the source sentence, so if the first word is

	Ylilauta	Suomi24	Both
Accuracy:	87.82	90.24	88.90
ERR:	-11.03	-6.08	-9.04
WER:	12.37	10.92	11.72
OR:	5.57	5.14	3.93
UR:	6.72	3.74	4.87

Table 5.8: Results of BRNN Mixed *

capitalized in the source, the first word is capitalized in the prediction too. The problem arises when the source sentence is not capitalized. This causes the model to not capitalize the first word in the prediction either.

	Original	Prediction	Target
1	Jos	Jos	Jos
2	tämä	tämä	tämä
3	toimii	toimii	toimii
4	tarjoan	tarjoan	tarjoan
5	sulle	sinulle	sinulle
6	kaljat	kaljat	kaljat
7	.	.	.

Table 5.9: An example of a sentence normalized by BRNN Mixed. Prediction errors are in bold.

5.2.4 BRNN Mixed Small

	Ylilauta	Suomi24	Both
Accuracy:	87.31	90.06	88.54
ERR:	-15.64	-7.98	-12.56
WER:	12.85	11.13	12.08
OR:	4.67	3.88	4.32
UR:	6.13	3.50	4.96

Table 5.10: Results of BRNN Mixed Small *

Table 5.10 shows that using less SSF data does not have a huge impact on the performance of the model. All of the evaluation scores of BRNN Mixed Small are slightly worse than those of BRNN Mixed, but the difference is quite small.

Table 5.11 shows and example of a sentence normalized by BRNN Mixed Small. According to the OR and UR, the model tends to under-normalize rather than over-normalize, but in this

case the model has attempted to normalize the word *julkisteta* (publish) into *julkistetaan*. This is grammatically incorrect in this context.

	Original	**Prediction**	**Target**
1	Tuloksia	Tuloksia	Tuloksia
2	ei	ei	ei
3	julkisteta	**julkistetaan**	julkisteta
4	kun	kun	kun
5	tutkinta	tutkinta	tutkinta
6	on	on	on
7	kesken	kesken	kesken
8	.	.	.

Table 5.11: An example of a sentence normalized by BRNN Mixed Small. Prediction errors are in bold.

5.2.5 BRNN Synthetic

The goal of the experiment with BRNN Synthetic was to see whether generating synthetic SSF data would improve the performance of the model. Table 5.12 shows that the performance of BRNN Synthetic is not significantly better than the other setups and it is actually slightly worse than BRNN Mixed and BRNN Mixed Small.

This could suggest that the noise-generation is not accurate enough to produce synthetic data that would improve the performance of the model significantly, even if data from a more appropriate domain is used.

This normalizer produced one error that no other normalizer in this study did. Table 5.13 shows that on line 10 the word *mutta* (but) has been capitalized even though this should not have been done. A closer look at the results shows that this is an isolated incident and that words after a comma or the word *mutta* are not incorrectly capitalized in any other case by the normalizer.

Generally, the mistakes made by BRNN Synthetic are quite similar to the errors by the other BRNN models, especially BRNN Mixed and BRNN Mixed Small.

5.2.6 Discussion of the BRNN models

Overall, many of the BRNN models suffer from the same problems. The models trained on less data struggle with more problems than the ones trained on more data, but the problems of the best models are also problems for the worse performing models.

Table 5.16 shows an example of the capitalization problems with the BRNN models. None of the BRNN models capitalize the first word, even though that is expected. All of the models

	Ylilauta	**Suomi24**	**Both**
Accuracy:	89.64	87.37	88.38
ERR:	-12.55	-15.13	-14.09
WER:	12.85	11.65	12.32
OR:	4.14	3.60	3.90
UR:	6.84	3.74	5.46

Table 5.12: Results of the BRNN Synthetic model *

	Original	**Prediction**	**Target**
1	Jos	Jos	Jos
2	oikein	oikein	oikein
3	olen	olen	olen
4	ymmärtänyt	ymmärtänyt	ymmärtänyt
5	niin	niin	niin
6	päätoimittaja	päätoimittaja	päätoimittaja
7	on	on	on
8	mies	mies	mies
9	,	,	,
10	mutta	**Mutta**	mutta
11	ei	ei	ei
12	ilmeisesti	ilmeisesti	ilmeisesti
13	mies	mies	mies
14	tuossakaan	tuossakaan	tuossakaan
15	talossa	talossa	talossa
16	.	.	.

Table 5.13: An example of a sentence normalized by the BRNN Synthetic -model. Prediction errors are in bold.

also struggle with the interjection *hei* (hey). Not surprisingly, the models trained with most data, i.e., BRNN Wiki and BRNN Mixed, perform the best. BRNN Synthetic also just normalizes *hei* incorrectly, but the rest of the sentence is left unnormalized as expected.

None of the BRNN models handle URLs well, just like the Murre normalizer. I use the same approach of copying the URLs from the source data to the predictions as I used with the Murre normalizer.

All of the BRNN models have trouble distinguishing between the two possible normalizations of the word *ku*. The possible normalizations are *kuin* (as/than/like) and *kun* (when). In table 5.15 there are two instances of this word, both in the meaning "when" on lines 7 and 43. Both of the models normalize the first instance incorrectly, but the model trained on more data normalizes the second instance correctly. This could show that the model trained on less data has not learned the other possible normalization at all, while the other model has. A deeper look

into the results shows that this is correct. The model trained on less data often normalizes the word *kun* as *kuin* even when that is not needed. The other model also makes similar mistakes but not to the same extent. This could suggest that the chunk size of 3 is not sufficient context for the models to learn which normalization is correct.

A notable problem with all the models with SSF data in the training set are also the clear differences between the normalization guidelines used for normalizing the SSF data and my own guidelines for normalizing the test set. One clear difference that comes up several times in the normalizations is that in the SSF data the phrases *ettei* (lest 3rd sg.) and *etten* (lest 1st sg.) are normalized as *että ei* and *että en* respectively. This is not wrong, but it is not needed because *ettei* and *etten* are also considered standard Finnish.

Table 5.14 shows one advantage of using the SSF data. The models trained without SSF data are not able to correct the dialectal expression *ootko* (are you) into the standardized version *oletko*. Both of the models trained also on SSF data however normalize this word correctly. This shows that using SSF data during training helps the models to learn how to handle these types of dialectal expressions.

Table 5.14 shows how BRNN Wiki Small and BRNN Mixed Small actually deal better with the dialectal *hakee* (pick someone up) on line 7. The three other models do not normalize this correctly and instead leave the word unnormalized. Most surprisingly, BRNN Wiki Small performs better than BRNN Wiki when both of the models are trained on data from the same source, and BRNN Wiki performs better on most other cases.

None of the BRNN models perform very well, but the BRNN Wiki model reaches a significantly better ERR than BRNN Wiki Small. Despite this, even the better model still got a negative ERR, which means that the model does not properly normalize the text.

	Source	BRNN Wiki	BRNN Wiki Small	Target
1	Ootko	**Ootko**	Oletko	Oletko
2	vielä	vielä	vielä	vielä
3	mestoilla	mestoilla	mestoilla	mestoilla
4	,	,	,	,
5	jos	jos	jos	jos
6	tullaan	**tullaan**	**tullaan**	tulemme
7	hakee	**hakee**	hakemaan	hakemaan
8	?	?	?	?

	BRNN Synthetic	BRNN Mixed	BRNN Mixed Small	Target
1	**Ootko**	**Ootko**	Oletko	Oletko
2	vielä	vielä	vielä	vielä
3	mestoilla	mestoilla	mestoilla	mestoilla
4	,	,	,	,
5	jos	jos	jos	jos
6	**tullaan**	**tullaan**	**tullaan**	tulemme
7	**hakee**	**hakee**	hakemaan	hakemaan
8	?	?	?	?

Table 5.14: An example of a sentence normalized the BRNN models.

	Source	BRNN Wiki	BRNN Wiki Small	Target
1	Ei	**ei**	**ei**	Ei
2	tee	tee	tee	tee
3	ellasta	**ellaista**	**elävasta**	Ellasta
4	mitenkää	mitenkään	mitenkään	mitenkään
5	kovaa	kovaa	kovaa	kovaa
6	kaveria	kaveria	kaveria	kaveria
7	ku	**kuin**	**kuin**	kun
8	pyörii	pyörii	pyörii	pyörii
9	uhkailemas	uhkailemassa	**uhkailemas**	uhkailemassa
10	MUIJIA	muijia	**Muijia**	muijia
11	kioskeis	kioskeissa	**kioskeis**	kioskeissa
12	ja	ja	ja	ja
13	kaupois	kaupoissa	kaupoissa	kaupoissa
14	!!	**B!**	**|½**	!!
15	!	!	!	!
16	Buahhahhaaa	**Buohhahhaa**	**Buahaa**	buahhahhaaa
17	!	!	**>**	!
18	!	!	**|**	!
19	!	!	**µ**	!
20	!	!	**>**	!
21	!	!	**|**	!
22	!	!	**µ**	!
23	!	!	**>**	!
24	!	!	**|**	!
25	!!	!!	**F½**	!!
26	!	!	**>**	!
27	!	!	**|**	!
28	!	!	**µ**	!
29	!	!	**>**	!
30	!	!	**|**	!
31	!	!	**µ**	!
32	!	!	**>**	!
33	!	!	**|**	!
34	!	!	**µ**	!
35	!	!	**>**	!
36	!	!	**|**	!
37	=D	=D	**ND**	=D
38	=D	=D	=D	=D
39	Sit	**Si**t	**Si**t	sitten
40	luikkii	**luikkia**	**luikkiin**	luikkii
41	pikkumies	pikkumies	pikkumies	pikkumies
42	pakoon	pakoon	pakoon	pakoon
43	ku	kun	**kuin**	kun
44	Securitas	Securitas	Securitas	Securitas
45	saapuu	saapuu	saapuu	saapuu
46	paikalle	paikalle	paikalle	paikalle
47	=D	=D	=D	=D
48	=D	=D	**SD**	=D
49	=D	**HD**	**ND**	=D

Table 5.15: An example of a sentence normalized by the two BRNN normalizers trained only on Wikipedia data. The tokenization is from the corpus.

	Source	BRNN Wiki	BRNN Wiki Small	Target
1	hei	**he**	**heim**	Hei
2	,	,	,	,
3	lahjoittakaa	lahjoittakaa	lahjoittakaa	lahjoittakaa
4	aivonne	aivonne	**aivonnet**	aivonne
5	tieteelle	tieteelle	tieteelle	tieteelle
6	,	,	,	,
7	heti	heti	heti	heti
8	huomenna	huomenna	huomenna	huomenna
9	!	!	!	!

	BRNN Synthetic	BRNN Mixed	BRNN Mixed Small	Target
1	**heidän**	**he**	**heidän**	Hei
2	,	,	,	,
3	lahjoittakaa	lahjoittakaa	**lahjoittakaan**	lahjoittakaa
4	aivonne	aivonne	aivonne	aivonne
5	tieteelle	tieteelle	tieteelle	tieteelle
6	,	,	,	,
7	heti	heti	heti	heti
8	huomenna	huomenna	huomenna	huomenna
9	!	!	!	!

Table 5.16: An example of a sentence normalized by the BRNN models

5.3 ByT5 results

5.3.1 ByT5 SSF

	Ylilauta	Suomi24	Both
Accuracy:	84.73	89.15	86.70
ERR:	-7.31	18.64	3.83
WER:	16.04	11.95	14.00
OR:	7.74	4.48	6.28
UR:	4.53	3.88	4.24

Table 5.17: Results of the ByT5 SSF model †

Table 5.17 shows that the first experiment with the ByT5 model provided notably better results than the previously discussed experiments with BRNNs. While the results are better, they are still not very good. The ERR score on the Suomi24 data is significantly better than on the Ylilauta data and thus this is the first model that has an overall ERR that is positive.

The best models by Samuel and Straka (2021) manage to gain an average ERR of 67.3% for all of the languages they developed ByT5 models for, while my model reaches an ERR of 3.83. The model they developed by only using the fine-tuning step also reached better results than my model, with an average ERR of 59.2%.

	Source	Murre	ByT5	Target
1	<URL>	<URL>	<URL>	<URL>
2	Mutta	mutta	mutta	mutta
3	ethän	ethän	ethän	ethän
4	sinä	sinä	sinä	sinä
5	euforinen	**euvorinen**	euforinen	euforinen
6	kaulaparta	kaulaparta	kaulaparta	kaulaparta
7	moiseen	moiseen	moiseen	moiseen
8	paskaan	**passkaan**	paskaan	paskaan
9	tietenkään	tietenkään	tietenkään	tietenkään
10	tuhlaa	tuhlaa	tuhlaa	tuhlaa
11	aikaasi	**aikaisin**	**aikaisin**	aikaasi
12	.	.	.	.

Table 5.18: An example of a sentence normalized by the Murre normalizer and the first ByT5 model. Prediction errors are in bold.

The OR of the model shows that training the ByT5 model on only dialectal data causes the model to over-normalize the less dialectal text from the Ylilauta and Suomi24 corpora. Newer slang words and English words are categorically over-normalized. Samuel and Straka (2021)

show that the pre-training step with synthetic data has a major effect on the results, so the results of this first experiment are not surprising.

As seen on table 5.19, the dialectal *ootko* (are you) and *hakee* (to pick up) are normalized correctly, but the slang word *mestoilla* (there/around) is incorrectly normalized as *metsoilla* (at/by the capercaillies). Despite getting slightly worse evaluation results, the Murre normalizer manages to normalize the sentence in this table correctly. The word *tullaan* is also left unnormalized which follows the Vilkuna (2014) guidelines but differs from the guidelines I followed during the manual normalization.

5.3.2 ByT5 Wiki

The second experiment provided clearly better results than the first one, as can be seen on table 5.20. The accuracy, ERR and WER all got much better than in the first experiment. Especially the ERR score on the Ylilauta corpus was improved. According to the ERR score, in the first ByT5 experiment the normalization of the Ylilauta corpus was worse than the LAI baseline. This experiment provided an ERR score of 13.33 on the corpus, which is noticeably on the positive side. These results are still far from the average ERR by Samuel and Straka (2021). The WER scores on the other hand are somewhat close to Partanen et al. (2019).

These results were somewhat affected by the normalization guidelines I decided to follow during the manual normalization. The problem was that the new training data did not have the beginnings of sentences in lowercase like the SSF data did. As I had followed the SSF guidelines in this regard, not lowercasing the first word of a sentence is considered "incorrect" and thus lowers all of the scores. To not have this difference affect the results, I uppercased the first words in the manually normalized sentences.

	Original	**ByT5 SSF**	**ByT5 Wiki**	**Target**
1	Ootko	oletko	**ootko**	oletko
2	vielä	vielä	vielä	vielä
3	mestoilla	**metsoilla**	mestoilla	mestoilla
4	,	,	,	,
5	jos	jos	jos	jos
6	tullaan	**tullaan**	**tullaan**	tulemme
7	hakee	hakemaan	**hakee**	hakemaan
8	?	?	?	?

Table 5.19: An example of a sentence normalized by the first and second ByT5 models. Prediction errors are in bold.

Table 5.21 shows that the model struggles with common dialectal expressions like *sun* (your) that should be normalized as *sinun*. This could be caused by the nature of the training data.

Wikipedia data does not generally have expressions like "mine" or "yours" so the model does not learn to normalize them. In the first experiment the model did learn to normalize words like *sun* correctly. The sentence in table 5.19 is normalized somewhat better by the first model than the second one, despite the overall performance of the second model being better. The words *ootko* (are you) and *hakee* (pick up) are not normalized by the second model, but the first one handles them correctly.

	Ylilauta	**Suomi24**	**Both**
Accuracy:	90.49	92.55	91.41
ERR:	13.33	19.01	15.62
WER:	9.73	7.73	8.84
OR:	1.72	1.26	1.51
UR:	6.44	4.72	5.68

Table 5.20: Results of the ByT5 Wiki model *

	Original	**Prediction**	**Target**
1	julkase	julkaise	julkaise
2	koodi	koodi	koodi
3	sitten	sitten	sitten
4	vaikka	vaikka	vaikka
5	githubiin	**githubiin**	Githubiin
6	niin	niin	niin
7	saadaan	saadaan	saadaan
3	karsittua	karsittua	karsittua
4	sun	**sun**	sinun
5	paskat	paskat	paskat
6	bugit	**bugit**	bugisi
7	pois	pois	pois

Table 5.21: An example of a sentence normalized by ByT5 Wiki. Prediction errors are in bold.

Compared to the best BRNN models, the ByT5 models handle unknown tokens and online-specific qualities of text better.

5.3.3 ByT5 Mixed

The third experiment with the ByT5 model provided worse results than the second experiment. Table 5.22 shows that the overall ERR score of the model is 11.79 while the second experiment reached the slightly better score of 15.62.

I was hoping to see if this model would handle spoken dialect in written form better than the previous model while also handling the special characteristics of online text better than the

first model trained only on SSF data. I originally thought that this would be possible, because the two BRNN models trained with a mix of synthetic Wikipedia data and SSF data learned to handle certain dialectal expressions better than the models solely trained on Wikipedia data.

	Ylilauta	**Suomi24**	**Both**
Accuracy:	89.79	92.55	91.02
ERR:	6.92	19.01	11.79
WER:	10.57	7.77	9.32
OR:	2.90	1.75	2.39
UR:	5.60	4.27	5.01

Table 5.22: Results of the ByT5 Mixed model*

5.3.4 Discussion of the ByT5 models

As seen in the previous sections, the ByT5 Wiki -model performs the best. The biggest problems with ByT5 Wiki and ByT5 Mixed is under-normalization, while ByT5 SSF tends to over-normalize. This means that using the synthetic data is clearly beneficial for the models, but the quality of the data could be better.

Other than the over-normalization by ByT5 SSF, there are no clear differences between the models. All of the models handle URLs very well and they also do not have problems with non-words.

Table 5.23 shows that all of the models correctly leave the emoticon ":D" on line 16 as it is. All of the models also manage to correct the typo on line 9 so *missääb* becomes *missään* (anywhere). Interestingly, ByT5 Wiki, the best of the three models, makes the only mistake in the sentence. It does not manage to normalize the word *olekkaan* on line 11 correctly into *olekaan* (be, present indicative connegative + suffix -kaan). The other two models do normalize this word correctly. This means that using SSF data in some form could help the models' performance.

5.4 Discussion

The experiments show that the ByT5 models tend to perform better than the BRNN models. ByT5 SSF performs worse than many of the BRNN models, but the two other ByT5 models are notably better than any of the BRNN models. There are some clear differences in the performances of the two types of models.

Firstly, the ByT5 models perform significantly better with URLs than the BRNN models. Table 5.18 shows a sentence normalized by Murre and ByT5 SSF. The <URL> tag marks the

	Original	ByT5 SSF	ByT5 Wiki	ByT5 Mixed	Target
1	Saattaa	saattaa	Saattaa	Saattaa	saattaa
2	tietty	tietty	tietty	tietty	tietty
3	olla	olla	olla	olla	olla
4	termistö	termistö	termistö	termistö	termistö
5	viturallaan	viturallaan	viturallaan	viturallaan	viturallaan
6	,	,	,	,	,
7	mutta	mutta	mutta	mutta	mutta
8	en	en	en	en	en
9	missääb	missään	missään	missään	missään
10	vaiheessa	vaiheesssa	vaiheessa	vaiheessa	vaiheessa
11	olekkaan	olekaan	**olekkaan**	olekaan	olekaan
12	väittänyt	väittänyt	väittänyt	väittänyt	väittänyt
13	olevani	olevani	olevani	olevani	olevani
14	sivistynyt	sivistynyt	sivistynyt	sivistynyt	sivistynyt
15	ihmisperse	ihmisperse	ihmisperse	ihmisperse	ihmisperse
16	:D	:D	:D	:D	:D

Table 5.23: An example of a sentence normalized by ByT5 Wiki. Prediction errors are in bold. Due to the different capitalization practises in the training data of the models, the capitalizations are not marked as errors.

URL "http://raapustus.net/?id=112". ByT5 SSF correctly leaves the URL unnormalized but the Murre normalizer changes the URL to "htsunraapustinnetustusinnetus.nettus.netaidalle". The post-processing script to correct URLs is not needed for the ByT5 normalizations. The difference between how well the ByT5 and the BRNN models handle URLs comes down to the pre-training of the original ByT5 foundation model. The SSF or the Wikipedia data do not contain any URLs so the BRNN models do not learn to handle them.

Other non-words are also handled better by the ByT5 model. The ByT5 models do not struggle with emoticons or the excessive use of punctuation marks like the BRNN models do in table 5.15.

The ByT5 models also do not have problems with unknown tokens in the same way as the BRNN models do. Table 5.24 shows the difference between the best ByT5 model, ByT5 Wiki, and the best BRNN model, BRNN Mixed. In the sentence in question, the writer has substituted the letter "ä" with the letter "à" on lines 2, 11 and 15. Neither one of the models is able to correctly substitute the "à" as an "ä", but the BRNN model clearly has more trouble handling the character. The ByT5 model just copies the character over, but the text remains more readable than in the BRNN model's normalization.

	Original	BRNN Mixed	ByT5 Wiki	Target
1	Kun	Kun	Kun	Kun
2	sydàn	**syd<unk>n**	**sydàn**	sydän
3	on	on	on	on
4	matkassa	matkassa	matkassa	matkassa
5	,	,	,	,
6	asiat	asiat	asiat	asiat
7	tuppaavat	tuppaavat	tuppaavat	tuppaavat
8	aina	aina	aina	aina
9	jollakin	jollakin	jollakin	jollakin
10	tavalla	tavalla	tavalla	tavalla
11	jàrjestymàän	**j<unk>rjestym<unk><unk>n**	**jàrjestymàän**	järjestymään
12	-	-	-	-
13	ellei	ellei	ellei	ellei
14	sitten	sitten	sitten	sitten
15	kyseessà	**kysees<unk>**	**kyseessà**	kyseessä
16	ole	ole	ole	ole
17	tahdon	tahdon	tahdon	tahdon
18	testaaminen	testaaminen	testaaminen	testaaminen
9	.	.	.	.

Table 5.24: An example of a sentence normalized by the best ByT5 and BRNN models.

There are also other cases where even the best BRNN model has to use the <unk> token. In many of these cases, the ByT5 models are able to normalize the character correctly or at least copy the character over. This is a big advantage for the ByT5 models, because all number of different characters can appear in social media text.

Chapter 6

Conclusions

The results of the first experiments clearly show that the Samples of Spoken Finnish data alone is not suitable for training models to normalize Finnish online forum texts. There are no existing tools suitable for normalizing Finnish social media text and further work is needed to get better results.

Suitable data for training tools for this task is also not readily available. Using Wikipedia data with artificial noise helped with certain normalization problems but also brought new ones.

The BRNN models are not a feasible option for normalizing Finnish social media text at this point of time. Based on Hämäläinen et al. (2020), BRNN models could work if better quality social media data was available. While we lack good quality training data, ByT5 based models are a better option. The ByT5 models experimented with in this thesis gained better results even without good quality normalized social media data.

The normalization guidelines followed in the manual normalization stage also clearly affect the results, as can be seen in the second experiment. Using data from different sources that follow different normalization guidelines can cause the models to learn something that in the evaluation stage is seen as incorrect although no mistake has actually been made. The problem is that there are no objectively "right" answers when it comes to normalization, so even "correct" normalizations can differ from each other and thus cause better or worse results than another way of normalization.

The results are somewhat affected by the differences between the training data sets. Because some of the results are evaluated against a test set that has been capitalized and some have been evaluated against a test set that has not been capitalized, not all of the results are totally comparable.

6.1 Further work

There is still a lot of work to be done when it comes to normalizing Finnish social media text. My experiments produced some promising results, even though they are far from the best results for the normalization of many other languages, like English.

Finnish normalization systems would greatly benefit from a manually annotated dataset of social media text. Annotating a larger amount of data from the Ylilauta and Suomi24 Sentences corpora could be a good starting point. Also improving the noise generation used to produce the synthetic Wikipedia data could be beneficial. Right now the approach is quite simple and the synthetic data could definitely be more similar to real world Finnish social media text.

Doing unsupervised pre-training for the ByT5 model on the Suomi24 and the Ylilauta data would likely generate better results. This would not require annotating more data by hand and would thus be easy to do. To pre-train the ByT5 foundation model on new data some of the data needs to be masked with sentinel tokens.

It would be beneficial to think about the guidelines used for the normalization and see which rules should be changed. For example, uppercasing could either not be used at all or in all the cases where it usually is used. This research is not focused on TTS or other spoken dialog systems, but doing the manual normalization in a way that could also be used to train these systems could also be considered. This would mean, for example, typing out numbers and abbreviations.

As Baldwin and Li (2015) point out, it is rare that research on lexical normalization examines the effects of the results on downstream tasks in detail. This limitation applies to this paper too and further experiments with downstream tasks are necessary to gain a better understanding of how useful the methods discussed in this study are in relation to downstream tasks.

Bibliography

Aller Media Ltd. The Suomi24 Corpus 2001-2017, VRT version 1.1, 2020. URL http://urn.fi/urn:nbn:fi:lb-2020021801.

AiTi Aw, Min Zhang, Juan Xiao, and Jian Su. A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 33–40, Sydney, Australia, July 2006. Association for Computational Linguistics. URL https://aclanthology.org/P06-2005.

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. How noisy social media text, how diffrnt social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 356–364, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I13-1041.

Tyler Baldwin and Yunyao Li. An in-depth analysis of the effect of text normalization in social media. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 420–429, 2015.

Anab Maulana Barik, Rahmad Mahendra, and Mirna Adriani. Normalization of indonesian-english code-mixed twitter data. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 417–424, 2019.

Marcel Bollmann. (Semi-) automatic normalization of historical texts using distance measures and the Norma tool. In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, pages 3–12, Lisbon, Portugal, 2012. Association for Computational Linguistics.

Marcel Bollmann. *Normalization of historical texts with neural network models*. PhD thesis, Ruhr University Bochum, Germany, 2018.

Marcel Bollmann. A large-scale comparison of historical text normalization systems. *arXiv preprint arXiv:1904.02036*, 2019.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Orphée De Clercq, Bart Desmet, Sarah Schulz, Els Lefever, and Véronique Hoste. Normalization of dutch user-generated content. In *9th International conference on Recent Advances in Natural Language Processing (RANLP 2013)*, pages 179–188. Incoma, 2013.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Peter Ebden and Richard Sproat. The kestrel tts text normalization system. *Natural Language Engineering*, 21(3):333–353, 2015.

Jacob Eisenstein. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `https://aclanthology.org/N13-1037`.

Hans Fix. Automatische Normalisierung - Vorarbeit zur Lemmatisierung eines diplomatischen altisländischen Textes. In Paul Sappler and Erich Straßner, editors, *Maschinelle Verarbeitung alt-deutscher Texte. Beiträge zum dritten Symposion Tübingen 17.-19. Februar 1977*, pages 92–100. Max Niemeyer Verlag, 1980.

Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar. Normalization of different Swedish dialects spoken in Finland. In *Proceedings of the 4th ACM SIGSPATIAL Workshop on Geospatial Humanities*, pages 24–27, 2020.

Institute for the Languages of Finland. Samples of Spoken Finnish, Downloadable Version, 2014. URL `http://urn.fi/urn:nbn:fi:lb-2020112937`.

Bryan Jurish. Comparing canonicalizations of historical german text. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 72–77, 2010.

Mike Kestemont, Walter Daelemans, and Guy De Pauw. Weigh your words—memory-based lemmatization for middle dutch. *Literary and Linguistic Computing*, 25(3):287–301, 2010.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `https://aclanthology.org/P17-4012`.

Krista Hannele Lagus, Minna Susanna Ruckenstein, Mika Pantzar, Marjoriikka Jelena Ylisiurua, et al. Suomi24: Muodonantoa aineistolle. 2016.

Lauta Media Ltd. Ylilauta, 2022a. URL https://ylilauta.org/.

Lauta Media Ltd. Ylilauta, 2022b. URL https://ylilauta.org/info/fi_fi/tietoa.

Lauta Media Ltd. Ylilauta, 2022c. URL https://ylilauta.org/statistics/.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Nikola Ljubešić, Katja Zupan, Darja Fišer, and Tomaž Erjavec. Normalising Slovene data: historical texts vs. user-generated content. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, pages 146–155, 2016.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL https://aclanthology.org/D15-1166.

Marian Macchi. Issues in text-to-speech synthesis. In *Proceedings. IEEE International Joint Symposia on Intelligence and Systems (Cat. No. 98EX174)*, pages 318–325. IEEE, 1998.

Courtney Mansfield, Ming Sun, Yuzong Liu, Ankur Gandhe, and Björn Hoffmeister. Neural text normalization with subword units. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 190–196, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-2024. URL https://aclanthology.org/N19-2024.

Robert Munro and Christopher D Manning. Short message communications: users, topics, and in-language processing. In *Proceedings of the 2nd ACM Symposium on Computing for Development*, pages 1–10, 2012.

Anna Björk Nikulásdóttir, Inga Rún Helgadóttir, Matthías Pétursson, and Jón Guðnason. Open asr for icelandic: Resources and a baseline system. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Niko Partanen, Mika Hämäläinen, and Khalid Alnajjar. Dialect text normalization to normative standard Finnish. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 141–146, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

David Samuel and Milan Straka. ÚFAL at MultiLexNorm 2021: Improving multilingual lexical normalization by fine-tuning ByT5. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 483–492, Online, November 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.wnut-1.54`.

Felipe Sánchez-Martínez, Isabel Martínez-Sempere, Xavier Ivars-Ribes, and Rafael C Carrasco. An open diachronic corpus of historical spanish: annotation criteria and automatic modernisation of spelling. *arXiv preprint arXiv:1306.3692*, 2013.

Yves Scherrer and Tomaž Erjavec. Modernizing historical slovene words with character-based smt. In *BSNLP 2013-4th Biennial Workshop on Balto-Slavic Natural Language Processing*, 2013.

Yves Scherrer and Nikola Ljubešić. Automatic normalisation of the Swiss German ArchiMob corpus using character-level machine translation. In *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, pages 248–255, 2016.

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Helga Svala Sigurðardóttir, Anna Björk Nikulásdóttir, and Jón Guðnason. Creating data in icelandic for text normalization. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 404–412, 2021.

Shubhi Tyagi, Antonio Bonafonte, Jaime Lorenzo-Trueba, and Javier Latorre. Proteno: Text normalization with limited data for fast deployment in text to speech systems. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 72–79, June 2021. URL `https://www.aclweb.org/anthology/2021.naacl-industry.10`.

Rob van der Goot. Monoise: A multi-lingual and easy-to-use lexical normalization tool. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 201–206, 2019a.

Rob van der Goot, Rik van Noord, and Gertjan van Noord. A taxonomy for in-depth evaluation of normalization for user generated content. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1109.

Rob van der Goot, Alan Ramponi, Arkaitz Zubiaga, Barbara Plank, Benjamin Muller, Iñaki San Vicente Roncal, Nikola Ljubešić, Özlem Çetinoğlu, Rahmad Mahendra, Talha Çolakoğlu, Timothy Baldwin, Tommaso Caselli, and Wladimir Sidorenko. MultiLexNorm: A shared task on multilingual lexical normalization. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 493–509, Online, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.wnut-1.55.

Rob Matthijs van der Goot. *Normalization and parsing algorithms for uncertain input*. PhD thesis, University of Groningen, 2019b.

Claudia Matos Veliz, Orphée De Clercq, and Veronique Hoste. Is neural always better? smt versus nmt for dutch text normalization. *Expert Systems with Applications*, 170:114500, 2021.

Maria Vilkuna. Skn-aineiston yleiskielistys, 2014.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *arXiv preprint arXiv:2105.13626*, 2021a.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41.

Ylilauta. Ylilauta Corpus, 2015. URL http://urn.fi/urn:nbn:fi:lb-2015031802.

Ylilauta. The Downloadable Version of the Ylilauta Corpus, 2016. URL http://urn.fi/urn:nbn:fi:lb-2016101210.