university of groningen / faculty of arts

Implementing a neural machine translation system for a low-resource language pair: the case of Gronings–Dutch

MA thesis

Information Science

Rick Kosse

s3243508

August 9, 2019

ABSTRACT

Gronings is a dialect of Dutch spoken in the north-eastern part of The Netherlands. While closely-related to Dutch, Gronings differs mainly in grammar (word order, and pro-drop for second person pronouns). Hardly any language technology nor language resources are available for Gronings. In this work we entered the challenge to build a machine translation (MT) system for Dutch to Gronings and vice versa despite the scarcity of resources. We approached this task by first processing and aligning a parallel corpus for Dutch-Gronings of roughly 8,000 sentence pairs. With the use of the NLTK library, sentences were split, tokenized and punctuation was normalized. Then, the sentences in both sides of the corpus were aligned with Hunalign. This sentence aligner supports the use of a bilingual dictionary to help the program align. A dictionary from a user-based website about Gronings was used, and Hunalign showed that recall scores were higher with the use of a dictionary than without. In recent studies, neural networks have proven to be very effective for MT and therefore our MT systems are built under this paradigm. Since Gronings is a dialect of Dutch and large part of the vocabulary is shared, we investigated which techniques are more effective for neural machine translation (NMT). We wanted to see whether character-based MT is more preferable than word-based MT or if the use of sub-word units is more valuable. Our attentional sequence to sequence system and ensemble method showed that with a tiny parallel corpus of 8k sentences, iterative back translation in combination with different amounts of hybrid data, reasonable translations are possible. Although the high baseline (BLEU score of 18 and 15), we were able to surpass the baseline with an improvement of +12.99 (in-domain BLEU score) and +14.62 (out-domain BLEU score). For our Gronings to Dutch system we even achieved better results with improvements in BLEU score of +20.76 and +14.51. For practical reasons we implemented our best models within a web API. The system translates reasonably although the translations are often too much of a mix between Dutch and Gronings. For further research we recommend more hand annotated data.

CONTENTS

PREFACE

With this Thesis I finally close the long journey of my educational career. Started on VMBO in high school, I would never thought I am standing here where I am now. For this I want to thank all the teachers, professors and study counselors at the University of Groningen who lectured me and helped me achieve this knowledge. Throughout this thesis, I had many support of people who I would like to thank. First of all the data providers prof. dr. Goffe Jensma and prof. dr. Martijn Wieling without their data this MT system would not be possible. Second of all, I would like to thank my supervisors prof. dr. Johan Bos and dr. Antonio Toral. They supported me and gave me numerous amount of advice. Furthermore, I am very grateful for the opportunity to present this Thesis at CLIN 29. The poster presented is attached in a Github repository.

Rick Kosse

"I have no special talent. I am only passionately curious." Albert Einstein

1 INTRODUCTION

In this thesis we explored the field of machine translation (MT) and specifically the field of low-resource language in combination with neural machine translation. We set up a research project to introduce a novel approach to translate Dutch to Gronings and vice versa. There currently is no MT system for Gronings and this deprives people of the opportunity of learning the dialect. Since Gronings is mostly spoken in and around the Dutch province of Groningen, we choose Dutch as our second language. All the code used in this thesis can be found in a Github repository[1].

1.1 MACHINE TRANSLATION

Automatic or machine translation is perhaps one of the most challenging tasks given the fluidity of human language. Classically, rule-based systems were used for this task, which were replaced in the 1990s with statistical methods. More recently, deep neural network models achieve state-of-the-art results in a field that is aptly named neural machine translation (NMT). NMT is an end-to-end learning approach for automated translation, with the potential to overcome many of the weaknesses of conventional phrase-based translation systems.

NMT approaches are highly reliant on the availability of large amounts of data and are known to perform poorly in low-resource settings. Recent MT research using small amounts of parallel texts showed that it is possible to build viable MT systems for low-resource pairs (Post et al., 2012). However, these systems have been shown to suffer from low accuracy (incorrect translation) and low coverage (high out-of-vocabulary rates), due to insufficient training data.

Currently, new processing techniques are mainly applied in order to achieve better translations results. Commonly word for word translations are used. However, character for character translation and sub-word units are becoming increasingly more popular (Wu et al., 2016).

In this thesis we want to build a functional MT system and combine it with the most effective processing technique to find out to what extent MT can be used to translate while dealing with scarcity of resources.

[1] https://github.com/rickirini/Flask-app-for-translation.git

1.2 MACHINE TRANSLATION AND LOW-RESOURCES

Common low-resource languages, such as dialects, are known for their lack of parallel data. Distinctively, dialects have a shared vocabulary with the language they originally descended from, although there might be a big contrast in grammar and word order.

An example of such a dialect with no MT system, is the dialect Gronings. Gronings derives from Low Saxon and is mainly spoken in and around the province of Groningen in the Netherlands. Although there is no translation system for Dutch <-> Gronings yet, research has shown that it is possible to combine MT with low-resource. Recently, a Frisian (second language of the Netherlands) MT system was build on 44.500 sentences (Gompel et al., 2014).

In order to translate Gronings some challenges may have to be overcome. Regional differences for example may influence translation results when word ambiguity occurs in the training data. Distinctive for a dialect, is the scarcity of resources. In order to train a supervised MT system, parallel data is necessary. However, the data available is often not parallel nor digital. Therefore, firstly, we want to investigate whether it is possible to set up a translation machine to translate Dutch to Gronings (and vice versa), despite the scarcity of resources **(Q1)**. Secondly, we want to find out to what extent we can use NMT for this job **(Q2)**, as research in this area is currently lacking. Thirdly, we want to test which processing method achieves the best results **(Q3)**. Fourthly, we will look for any additional methods that could further enhance the translation score **(Q4)**. Finally, we aim to deploy our best model in a web API **(Q5)**.

In the following Section (2) of this thesis we start off with investigating the research which has already been conducted on MT and especially in combination with low-resource languages and several processing methods. In Section 3 the collected corpora will be described along with a brief summary of the characteristics of Gronings. After the final collection, we describe the processing methods which are applied to the data to make it NMT ready (Section 4). After conducting our experiments we put our baseline score next to our results with different processing methods in order to find out which combination performs best. Then we describe how we implemented our final model within a user-friendly web API and look at the given translations next to the reference sentence to give a visual image of the results (Section 5).

2 PREVIOUS WORK ON MT FOR LOW-RESOURCE LANGUAGES

In order to answer our research questions, we first need to look at previous work on MT in combination with low-resource languages. In this chapter we describe research conducted on NMT, MT in combination with low-resources, preprocessing, encoding, backtranslation methods and ensembling models.

2.1 LOW-RESOURCES IN COMBINATION WITH MACHINE TRANSLATION

NMT systems are highly reliant on large amounts of data and are known to perform poorly in low-resource settings. Unfortunately, languages such as Gronings suffer from the fact that there is not much parallel data available. Gompel et al. (2014) proposed an statistical machine translation (SMT) model (created with Moses) for the low-resource language of Frisian. Their NL $\leftrightarrow$ FRI model, trained on 44.500 sentences, scored around 51 Bilingual Evaluation Understudy (BLEU) points. Otte et al. (2011) described a rule-based approach for machine translation between Dutch and Afrikaans. Their system relied heavily on the re-use of publically available resources such as Wiktionary, Wikipedia and the Apertium machine translation platform (Forcada et al., 2011). In languages which have a shared vocabulary, Unhammer and Trosterud (2009) described the development of a two-way shallow-transfer machine translation system between Norwegian Nynorsk and Norwegian Bokmal. It was built on the Apertium platform, using the open source resources Norsk Ordbank and the Oslo–Bergen Constraint Grammar tagger. Their machine translation system seemed to work well for languages with shared vocabulary, because a BLEU score of 74 was achieved.

2.2 NEURAL MACHINE TRANSLATION

After decades of SMT, in 1997 Neco and Forcada (1997) showed the first forms of recurrent neural networks (RNN). Later, in 2013, Kalchbrenner and Blunsom proposed a new end-to-end encoder-decoder structure for machine translation (Blunsom and Grefenstette, 2013). This model encodes a given source text into a continu-

ous vector using convolutional neutral network (CNN), and then uses RNN as the decoder to transform the state vector into the target language. Their work can be seen as the birth of NMT, which is a method that uses deep learning neural networks to map among natural language. One year later (2014), Sutskever et al. (2014) and Cho et al. (2014) developed a method called sequence to sequence (seq2seq) learning using RNN for both encoder and decoder. The power of this model is that it can map sequences of different lengths to each other.

In 2014, Bahdanau et al. (2014) introduced the "attention" mechanism to NMT. Their attention mechanism enabled the neural network to focus more on the relevant parts than the irrelevant parts when doing a prediction task (Bahdanau et al., 2014). When the decoder is generating a word to form the target sentence, only a small portion of the source sentence is relevant; thus a content-based attention mechanism is applied to generate a context based vector on the source sentence. The target word will then be predicted based on the context vectors instead of a fixed-length vector.

Recently, the concept of 'attention' became increasingly popular in training neural networks, allowing models to learn alignments between different modalities. In the context of NMT, Bahdanau et al. (2014) showed that NMT combined with attentional mechanisms can translate and align words successfully. The performance of NMT was dramatically improved due to the rise of the attentional encoder-decoder networks. Nowadays, it has become the state of the art in the field of NMT (Vaswani et al., 2017).

One of the most used NMT systems is the NMT-Keras system of Álvaro Peris and Casacuberta (2018). This NMT system provides a user friendly attentional RNN mechanism. The system is meant for training, decoding and scoring translation models. NMT-Keras is based on an extended version of the popular Keras library (Chollet et al., 2015) and it runs atop of Theano (Bergstra et al., 2010) or Tensorflow (Abadi et al., 2016).

2.3 PARALLEL DATA

In order to retrieve predictions from a MT system, parallel data is needed for training, development and testing. Parallel data is textual data of two different languages which are aligned at the sentence level. Toral et al. (2012) describes a novel efficiency-based evaluation of sentence- and word aligners. Among these sentence aligners were Hunalign (Varga et al., 2007), GMA1 and BSA (Moore, 2002). In the research of Pecina et al. (2011) Hunalign was used to align their sentences for SMT. Hunalign is a widely used tool for automatic identification of parallel sentences in

parallel texts. Hunalign also provides a score which reflects the level of parallelness, the degree to which the sentences are mutually aligned (Varga et al., 2007).

2.4 WORD VS CHARACTER VS BPE

The use of character NMT is rising in popularity, followed by Costa-Jussa and Fonollosa (2016) and Kim et al. (2016). Their systems used character-based (char-based) embeddings in combination with convolutional and highway layers to replace the standard lookup-based word representations. They obtained three BLEU points more for their German-English system than their baseline auto-encoder architecture with an attention-based mechanism. Another char-based model was developed by Pettersson et al. (2013). They proposed an approach with tagging and parsing of historical text, using char-based SMT methods for translating the historical spelling to a modern spelling. They showed that their approach for spelling normalization is successful even with small amounts of training data. van Noord et al. (2018) used four encoding models to produce Discourse Representation Structures (DRS) for English sentences. The presented models were a word-based model, a char-based model, a hybrid representation of subword units (BPE) and a combined characters and word model. The results showed that char-based model outperformed the word-based model. The hybrid representation is a model originally from Sennrich et al. (2015b). In NMT, BPE is a frequency-based method that automatically finds a representation that is in between character and word-level. It starts out with the character level format and then does a predefined number of merges of frequently co-occurring characters. Tuning this number of merges determines if the resulting representation is closer to character- or word-level. It is for example used in the research of Vaswani et al. (2017) who proposed a novel, yet simple network architecture based solely on an attention mechanism dispensing with recurrence and convolutions entirely. Their English to French system, outperformed the previous single state-of-the-art model by 0.7 BLEU points, achieving a BLEU score of 41.1.

2.5 BACKTRANSLATION

When only small datasets are available, as in this study, the technique of backtranslation is often applied to create a bigger (synthetic) datasets (Sennrich et al., 2015a). Recently, researchers have shown that backtranslating monolingual data can be used to create synthetic parallel corpora, which can be merged with human-translated (or authentic) data to train a high-quality NMT system. Poncelas et al.

(2018) used incrementally large amounts of backtranslated data to train a range of NMT systems for German-to-English. Their first system was trained on 1M sentences of authentic parallel data, the second system was build using solely synthetic data. The synthetic data system achieved a higher BLEU score than the authentic data system (22.9 BLEU points on 1M synthetic sentences and 22.78 BLEU points on 1M authentic sentences respectively). Next they combined an initial set of authentic data with the synthetic data, a so called hybrid model. They analyzed the hybrid NMT models and showed that while translation performance tends to improve when larger amounts of synthetic data are added, performance appears to tail off when the balance is tipped too far in favour of the synthetic data.

Building further on backtranslation, the research of Hoang et al. (2018) introduced iterative backtranslation. Their simple yet effective method showed that with backtranslated data they were able to build a translation system in forward and backward directions, which in turn is used to re-backtranslate monolingual data. This process can be "iterated" several times . A visual explanation is shown in Figure 1.

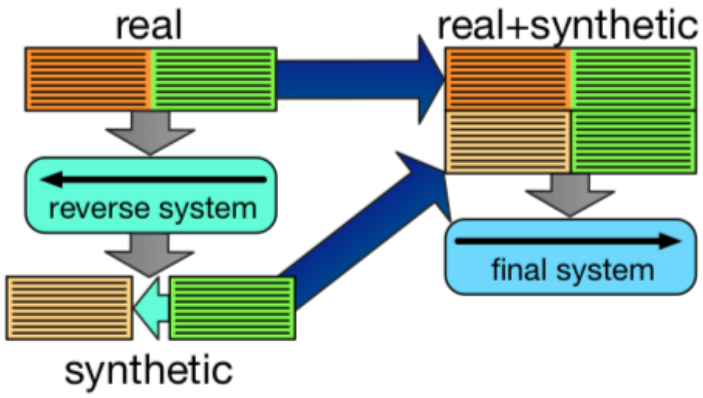

Figure 1: The process of iterative backtranslation (Hoang et al., 2018)

The more iterations, the better the data got translated and thus better results were achieved in the end by the system. Hoang et al. (2018) showed that for their Farsi–English MT system in low-resource settings (100k sentences) they were able to surpass the NMT baseline system with 0.9 BLEU points after two iterations. For the direction English-Farsi they achieved an improvement of 0.6 BLEU points compared to the NMT baseline without iterative backtranslation.

2.6 ENSEMBLING

One of the additional features of the system of Álvaro Peris and Casacuberta (2018) is the ensemble method. At each step during sequence prediction, a translation system outputs a full probability distribution over the target vocabulary. Therefore, the task of NMT system combination can be cast as an ensemble prediction task and a variety of existing general prediction combination methods can be applied. This method has been used in word-based and char-based models by Ling et al. (2015) and has shown to be effective. Luong et al. (2015) ensembled eight NMT models in order to get better translations. They showed that they achieved one BLEU score more with the ensemble method.

3 PARALLEL DATA FOR GRONINGS

In this chapter we look at the characteristics of Gronings along with the collection of data needed for our MT system. The parallel data obtained was mainly in Dutch and Gronings. Therefore, it was decided that our system should translate between Dutch and Gronings. In this Section we describe the characteristics of Gronings, the final collection of the data and we describe several methods in order to prepare the data for alignment.

3.1 GRONINGS

As described in Section 1.2, Gronings is a regional language spoken in and around the Dutch province of Groningen and is derived from Low Saxon. At the beginning of the 20th century, Gronings was still the most important language in and around the province of Groningen. Nowadays only a small part (558,000[1]) of the people in Groningen speak the dialect, even though it is in increasingly more Dutch forms. That proportion is even smaller among younger generations. Many parents choose to raise their children in Dutch in order to prevent alleged arrears. Also Groninger immigrants have become numerous and do not take over Gronings in an active sense. This is at the expense of the language itself. Despite the shared-vocabulary with Dutch, Gronings is characterized by its own vocabulary and word order, which differs greatly from the other Low Saxon dialects.

3.1.1 Pronoun drop

Just like in Frisian (the second language of the Netherlands), the word doe (you) can often be omitted. For every word that comes before doe, ends with '-s', so you already know that the word doe stands behind it. When doe is used, emphasis is placed on this. Examples are given in Table 1.

[1] http://taal.phileon.nl/nds_gronings.php

Table 1: Examples of pro drop in Gronings compared to Dutch (Wikipedia-contributors, 2019)

Language	Sentence
Gronings	Hest dat doan?
Dutch	Heb je dat (even) gedaan?
Gronings	Hestoe dat doan?!
Dutch	Heb jij dat gedaan?!

3.1.2 Verbs

Another distinct characteristic of Gronings, regarding verbs, is that there are many verbs in Gronings which are not used in Dutch. For example the conjugations, the past participle and certain forms of verbs are different. For regular verbs, the suffixes are the same with irregular verbs, but the stem can vary. In the hai form, the -t can be omitted in some dialects. When the stem ends in an -l or an -r, only one -n is left behind in the plural forms, so not fiedelen but fiedeln or in Dutch viool spelen (violin playing). Overall, Gronings has many irregular verbs, many more than there are in Dutch.

3.1.3 Word order

A feature of Gronings that shows its relationship with Frisian and the distance to Dutch is the word order. In some cases this does not correspond with Dutch and is often regarded as a substandard for non-speakers of Gronings. This word-order also occurs in Frisian. For example the auxiliary verb comes in the end of the sentence. An example is given in Table 2.

Table 2: Examples of word order in Gronings and Dutch (Wikipedia-contributors, 2019)

Language	Sentence
Gronings	Zeg mor davve nai' kommen willen.
Dutch	Zeg maar dat wij niet willen komen.

3.1.4 Regional differences

There are many varieties of Gronings within the province of Groningen as seen in Figure 2. These regional differences are often referred to by the name of the place or region in which they are spoken. The following dialects can be distinguished:

the Kollumerpompsters, the Stadjeders, the Hogelandsters, the Oldambtsters, the Veenkoloniaals and the Westerwolds.

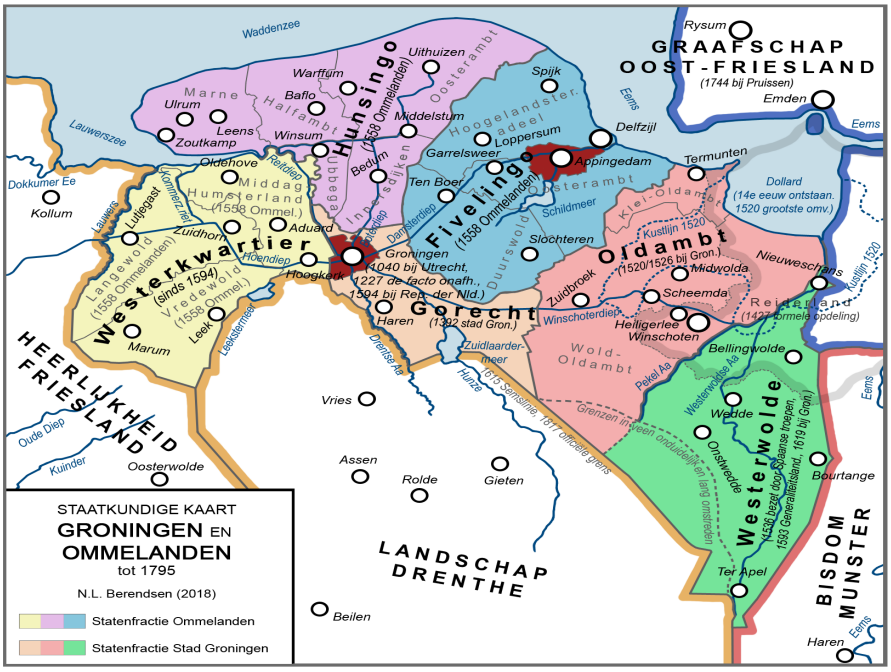

Figure 2: The province of Groningen and its regional differences in Gronings; Kollumerpompsters (yellow), the Stadjeders (center red), the Hogelandsters (purple), the Oldambtsters(light red), the Veenkoloniaals (North Drenthe) and the Westerwolds (green) (Berendsen, 2019)

3.2 DATA COLLECTION

As described in one of the challenges in Section 2.1, there was not much parallel literature available. If there was a parallel corpus available, the literature was often not digital. Our search for literature yielded two different corpora. One of the corpora is the book Martha (Visscher, 2004) about the second world war. Therefore, we found various anti-Semitic sentences in the corpus which might end up in our MT system. Nevertheless, the corpus contains around 1200 parallel sentences. The second corpus found was Goud Volk (GV) (Reker, 2008). This five-part book was written by Siemon Reker. Reker is professor in language and culture of Gronings[2]. Goud Volk contains different sorts of literature such as poems, plays and short stories. In total, this book contains around 8,000 parallel sentences. Considering the book is written by a professor in language and culture of Gronings, the writing style of this book can be seen as a standard in flawless Gronings.

Futhermore, additional Groninger monolingual data was found. For example a book with the fairy tales of the brothers Grimm (Dijken, 2016). Unfortunately,

[2] https://www.rug.nl/news/2007/10/108_07

the author translated the book from German to Gronings. This means that the Groninger version of the Grimm does not correspond to its Dutch version and is therefore not useful for alignment. Other Groninger monolingual books found are the Groninger Biebel (Liudgerstichten, 2007) (Bible) and Children bible (Binsbergen e.a, 2008). The Biebel is translated from Hebrew and does not match with the Dutch version. Children Bible was not available in Dutch. Nevertheless, these three monolingual books may come useful when it comes to backtranslation, which is described in Section 2.5. For Dutch we found a monolingual transcribed Ted talk[3] corpus consisting of 80,000 Dutch sentences. An overview of the literature including the amount of sentences and tokens (words) per corpus is shown in Table 3. An example of parallel sentences, retrieved from GV and Martha, are given in Table 4.

Table 3: Overview of the corpora obtained; sent = amount of sentences, tokens = amount of words, k = thousand, M = million

Corpus	Author	Parallel	Sort	Sent	Tokens
Goud Volk	S.Reker	✓	Poems	8k	105k
Martha	K.Visscher	✓	Story	1,2k	40k
Grimm	M. van Dijken	Gronings	Story	900	27k
Bible	Liudgerstichten	Gronings	Bible	40k	578k
Children bible	M. van Dijken	Gronings	Bible	800	17k
Ted Corpus	unknown	Dutch	Transcribed talks	448k	3M

Table 4: Parallel sentences of the Goud Volk and Martha corpora

Corpus	Dutch	Gronings
GV	Dat is toch zeker niet waar, dominee?	't is doch wizze nich waôr, doomdie?
	En moeder laat haar gaan al is het met groot verdriet.	En moeke let 'eur loop'n al is 't mit groot verdrait.
	Vaak had hij een nare droom, dan snurkte hij en riep.	Voak had hai ook 'n noare dreum den snurkte en den ruip hai.
Martha	Mijn moeder veegde haar schuimhanden af aan haar bonte schort.	Mien moe veegde heur schoemhanden of aan heur bonde schoet.
	Jan knien is er niet meer.	Jan knien is der nait meer.
	Een breed rood lint hield haar krullen bij elkaar.	n braide rooie lint huil heur krullen bienander.
	Ik stond verloren in de berm van de gracht.	Ik ston verloren op d'wiekswale.

3.3 CONVERTING AND CLEANING

After converting GV from PDF to txt file with a web based PDF to txt tool[4], the corpus contained some irrelevant text. Cases of time stamps, introduction and

[3] https://github.com/ajinkyakulkarni14/TED-Multilingual-Parallel-Corpus/tree/master/Monolingual_data
[4] https://pdftotext.com/nl/

footnotes were included in the text. These were not parallel and thus removed by the use of Regex[5] commands. After the clean up, the final GV corpus remained with text and page numbers. The contents of the corpus were arranged as seen in Table 5.

Table 5: Goud Volk structure of contents after cleaning

section	language	page
Poem 1 (part 1) paragraph	Dutch	1
Poem 1 (part 1) paragraph	Gronings	2
Poem 1 (part 2) paragraph	Dutch	3
Poem 1 (part 2) paragraph	Gronings	4

As seen above, part 1 of the poems stopped at the end of the page and continued on the next page in the other language. Once on the following page, they continued the first poem (part 2) in the first language again.

3.4 PARAGRAPH SPLITTING

Because of the ordering of the paragraphs, we have placed <p> tags between the paragraphs to define the beginning and end of a poem. This way our aligning system had the possibility to keep track of the beginning and end of a poem so that the system knew which sentences belonged to which corresponding poem. An example is given in Table 6.

Table 6: Example of Goud Volk alignment

Section	part	language	align	section	part	language
Poem 1	(part 1)	Dutch	<aligned to>	Poem 1	(part 1)	Gronings
Poem 1	(part 2)	Dutch	<aligned to>	Poem 1	(part 2)	Gronings

After the alignment of the paragraphs, the next task was to align the sentences. However, before aligning, various methods of preprocessing have been applied to achieve the best alignment score possible. A small summary of the methods applied are described in the following Sections.

[5] https://regexr.com/

3.4.1 Sentence splitting

This is done with the Python3 NLTK library[6]. This library contains a sent_tokenize function which automatically detects sentences in a corpus and puts them on a newline. This was useful to distinguish quotes. Often these quotes were hard to detect by our own sentence splitting script (which splittted on punctuation) since the quotes contained improper use of punctuation and therefore were seen as a sentence on its own. An example is given in Table 7.

Table 7: Examples of sentence splitting on quotes

Function	**Sentence**
Reference	1: "Hest dat doan?" zee Martha!
Sentence splitting script	1: "Hest dat doan?
	2: "zee Martha!
NLTK sent_tokenize	1: "Hest dat doan?" zee Martha!

3.4.2 Excessive newlines

To match the original format of the book, the text contained lots of newline characters in the middle of the sentence. These excessive newlines were removed to set the sentences back to its original length.

3.4.3 Normalization and separating punctuation

Noticeable was the use of different punctuation within the corpus, e.g. the quotation marks. Sometimes the single quotation marks were used and sometimes double quotation marks. These punctuations had to be normalized to one sort of punctuation or had to be removed entirely. This was done with Moses normalize-punctuation[7]. Furthermore, every punctuation is separated from the word with a white space since the NMT system sees words with concatenated punctuation as a different word.

[6] https://www.nltk.org/

[7] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/normalize-punctuation.perl

3.4.4 Lower casing

Finally to improve alignment quality the sentences were lower cased with the Python build in function .lower()[8].

[8] https://www.programiz.com/python-programming/methods/string/lower

4 ALIGNMENT AND TRANSLATION METHODOLOGY

Once our corpora was collected and cleaned, the sentences were ready for alignment. In this chapter we describe the best method to align our sentences for the benefit of training our NMT system.

4.1 HUNALIGN AND PARALLELNESS OF ALIGNMENT

After preprocessing, the program Hunalign was used to get the sentence alignment done. This is an open source program which is especially useful since it has different parameters to reflect the parallelness. This measurement is given by recall score. Once the data was handed to Hunalign. Several parameters were set in order to obtain the best possible score. Furthermore, the realign and dictionary features of Hunalign were used.

If the realign feature is set, the alignment is built in three phases. After an initial alignment, the algorithm heuristically adds items to the dictionary based on co-occurrences in the identified sentence pairs. Then it re-runs the alignment process based on this larger dictionary. This option is recommended to achieve the highest possible alignment quality. It is not set by default because it approximately triples the running time.

In the presence of a dictionary, Hunalign uses it, combining this information with Gale-Church sentence-length information (Gale and Church, 1993). In the absence of a dictionary, it first falls back to sentence-length information, and then builds an automatic dictionary based on this alignment. Then it realigns the text in a second pass, using the automatic dictionary. Since there is no official bilingual Dutch-Gronings dictionary online, we tried two different kinds of bilingual dictionaries. The first is from a user based website[1]. The second bilingual dictionary is the GV corpus splitted and aligned on word level.

Furthermore, we noticed that Hunalign often performs better when quotation marks were removed. As described in Section 3.4.1 quotes were hard to detect due to improper use of punctuation. Therefore, we experimented with- and without the use of quotation marks in the text. The recall scores of the alignment along with the parameters are shown in Table 8.

[1] https://www.mijnwoordenboek.nl/dialect/Gronings

Table 8: Hunalign recall scores combined with/without quotation marks, different parameters and dictionaries; **Bold**= best alignment score

Model	Score	Dictionary	Parameter
With quotation marks	1.71763	null	-text -bisent
Without quotation marks	1.7067	null	-text, -bisent
With quotation marks	1.6124	null	-text,-bisent, -realign
Without quotation marks	1.65334	null	-text, -bisent , -realign
With quotation marks	**1.74192**	User dictionary	-text -bisent
With quotation marks	1.61147	User dictionary	-text -bisent -realign
Without quotation marks	1.73021	User dictionary	-text, -bisent
Without quotation marks	1.73021	User dictionary	-text, -bisent, -realign
With quotation marks	1.56748	GV dictionary	-text -bisent
Without quotation marks	1.55948	GV dictionary	-text, -bisent

As seen in Table 8 the highest recall score (1.74192) achieved is with quotation marks along with the user dictionary and without realigning. This method was applied to the GV and Martha corpora resulting in two txt files containing the alignment.

4.2 REVISING ALIGNMENT

Before training our NMT system, we inspected the sentence alignment given by Hunalign. Unfortunately, not all sentences of Martha were aligned as accurately as we had hoped. To make sure the alignments were correct, we hand annotated around 600 sentences. This was done with a Python 3 script. In Figure 3 an example is shown of how the script looks when executed.

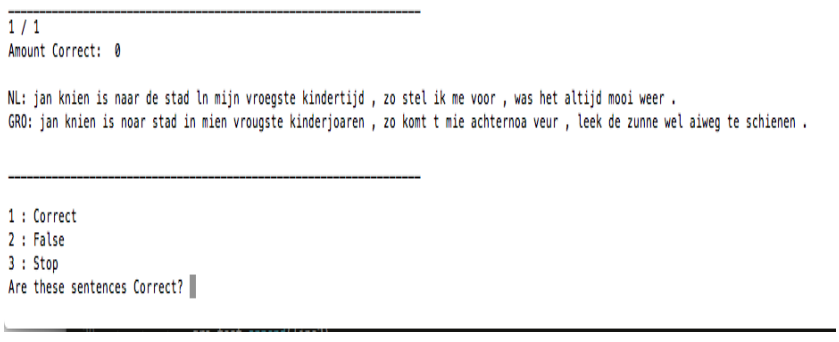

Figure 3: Python program used for hand annotating the Martha corpus

When a Dutch and Groninger sentence did align, the annotator pressed the corresponding number. The sentence was then written to a file accordingly. When

incorrect, the sentence was set aside. In the end, we collected a corpus of 517 hand annotated sentences. Since Martha's alignment was hand annotated and contained anti-Semitic sentences, we decided to leave the corpus out of the training data. The sentences from the Martha corpus, were used solely as out-domain test set. The GV corpus was used for train, development and test (in-domain) purposes.

4.3 TRAIN, DEVELOPMENT AND TEST

After aligning, the data was splitted into eight files named training (train), development (dev), in-domain test and out-domain test for Dutch and Gronings. This was distinguished by a .gro or .nl extension (e.g. train.nl or dev.gro). The file splitting was done with the function train_test_split from Sklearn[2]. First the train and test data were created by splitting the entire corpus with a ratio of 8/2. Here the shuffle parameter of the train_test_split function was set in order to shuffle the sentences. Next, the same method was used on the train files creating new train and dev files. This ended up with a ratio of 6/2/2 for train, test and dev. The final sets along with the amount of sentences and tokens are shown in Table 9.

Table 9: Train, test and dev sets for both languages along with the amount of sentences and tokens for each set

Set	language	Num sents	Num tokens
Train		6,681	94,540
Dev	GRO	464	7,008
Test-in		496	7,063
Test-out		515	6,659
Train		6,681	95,401
Dev	NL	464	7,003
Test-in		496	7,091
Test-out		515	6,799

4.4 CREATING THE CHARACTER ENCODING MODEL

After the files were created, we applied various encoding models. In essence, the word model is already created since our current sets consist out of words and the punctuation is already separated by a white space as described in Section 3.4.3. In

[2] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

order to create the character model, the word-based models had to be copied and transformed from words to characters separated by white spaces. The train, dev, and test files were taken from the word-based model to make sure the data was the same among all models. First, the original white spaces of the sentences were replaced with three star (***) symbols. This is a unique symbol which did not occur in the training data and was also applied in van Noord et al. (2018). Finally every character was split with a white space in between them. An example of the encoded data is given in Table 10.

4.5 CREATING THE BYTE PAIR ENCODING MODEL

Like the character model, the train, dev and test files were taken from the word-based model. The BPE encoding was applied following the guidelines of Sennrich et al. (2015b). First the three files (test out-domain sets left aside) were copied and put together to create a vocab file per language. This way the algorithm could count the N most frequent sequences, iteratively. Next, the train, dev, in-domain test and out-domain test files were processed with 50, 100, 250, 500, 750 and 1000 merges and written to a separate file per merge. The more merges the model contained the more byte pairs were encoded. The N most frequent sequences were replaced with an @ symbol. Examples of the BPE data is given in Table 10.

Table 10: Examples of the word, char, BPE-50 and BPE-1000 encoding along with the amount of tokens for the sentence

Model	Example sentences	Tokens
Word-based:	ze hadden het land geïnspecteerd en hij had haar de gewassen laten zien .	14
Char-based:	z e *** h a d d e n *** h e t *** l a n d *** g e ï n s p e c t e e r d *** e n *** h i j *** h a d *** h a a r *** d e *** g e w a s s e n *** l a t e n *** z i e n *** .	73
BPE-50:	"ze h@@ a@@ d@@ d@@ en het l@@ an@@ d ge@@ ï@@ n@@ s@@ p@@ e@@ c@@ t@@ e@@ er@@ d en h@@ ij h@@ a@@ d h@@ aar de ge@@ wa@@ s@@ s@@ en l@@ a@@ t@@ en z@@ ien ."	44
BPE-1000	"ze h@@ a@@ d@@ d@@ en het l@@ a@@ n@@ d ge@@ ï@@ n@@ s@@ p@@ e@@ c@@ t@@ e@@ e@@ r@@ d en hij h@@ a@@ d h@@ a@@ a@@ r de ge@@ w@@ a@@ s@@ s@@ en l@@ a@@ t@@ en z@@ i@@ en ."	23

4.6 RESTORATION AND EVALUATION

To make the evaluation possible, after translation, the character- and BPE output had to be set back to word-based sentences since we evaluated them relative to the preprocessed word-based test files. For the character model this is done by removing all white space and then replace each *** symbol by a new white space via a simple Regex command. For the BPE encoding a simple Linux Bash replacement command was used: sed -r 's/(@@)|(@@?$)//g'.

After restoring, the sentences needed to be evaluated. This is done with the Bilingual Evaluation Understudy Score (BLEU) (Papineni et al., 2002) as implemented in Moses[3]. This package returns the BLEU score, N-gram precision and brevity penalty of a given set. The BLEU score is a metric for evaluating a generated sentence to a reference sentence. A perfect match results in a score of 1.0, whereas a perfect mismatch results in a score of 0.0. BLEU score does not only count the word precision but also words that occur next to each other. These are called N-grams, where N is the number of words per group. Unigrams, bigrams, trigrams and four-grams consist of chunks of one, two, three and four words respectively. BLEU measures by looking at N-grams overlap between the output and reference translations with a penalty for shorter outputs. The penalty, known as brevity penalty, penalizes sentences that are shorter than any of the reference translations. We can do this by comparing the output sentence to the length of the reference sentence. If our output is as long or longer than any reference sentence, the penalty is 1. Furthermore, the metric looks at each word in the output sentence and assigns it a score of 1 if it shows up in any of the reference sentences and 0 if it does not. Finally, to normalize the count to an interval between 0 and 1, the metric can divide the number of words that showed up in one of the reference translations by the total number of words in the output sentence. This gives us a measure called N-gram precision.

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

5 EXPERIMENTS AND DISCUSSION

In this chapter the results of our experiments are described. First the performance and parameter settings of the base models (word-, character- and BPE encoding) for both directions are reviewed. Next we describe the results of our best base model in combination with monolingual (synthetic) data via backtranslation and iterative backtranslation. After that we try to enhance our score with the ensemble method. Since BLEU scores do not tell everything about the correctness, we provide a small analysis of the translations given by our best models. To make our system practical we implement our best model within a web API. Hereafter, we describe the implementation, flowchart, results and libraries used for the web API. Finally the discussion of this thesis will be given.

5.1 RESULTS BASE MODELS

Table 13 shows the results of our base models with word, character- and BPE- encoding. The N-gram (n=1-4) precision and brevity penalty are shown in Appendix A. The translations were evaluated on the in-domain (Goud Volk) and out-domain (Martha) test sets. Since Gronings and Dutch are closely related, first, we compared the test files without translating to see which scores (baseline score) we had to surpass as seen in the first row of Table 13.

5.1.1 Parameters

The system was trained with Keras atop of Tensorflow. It was trained on a Nvidia K40 GPU. Per model, the amount of epochs (term for when the dataset is passed forward and backward through the neural network once) are shown in Table 11. The hyperparameters used for all the models are shown in Table 12. The performance of the models were evaluated in BLEU scores as described in Section 4.6.

Table 11: Epochs per model per language

Model	NL→ GRO	GRO→ NL
word	32	34
character	51	91
BPE 50	42	58
BPE 100	67	53
BPE 250	74	68
BPE 500	66	55
BPE 750	54	50
BPE 1000	44	45

Table 12: Keras hyperparameters for all models

Hyperparameters	Value
POS_UNK	True
Loss	categorical
Activation	softmax
Sample_weights	True
Optimizer	Adam
Learning rate	0.001
Max_epoch	500
Batch_size	50
N_GPUS	1
Early_stop	True
Patience	15
Stop_metric	BLEU_4
Model_TYPE	Attention RNN
Encoder_RNN	LSTM
Decoder_RNN	Conditional LSTM
USE_CUDNN	True

Table 13: BLEU scores of the base models: word, character and BPE encoding; **Bold** = best score obtained

Model	NL-GRO		GRO-NL	
	In	**Out**	**In**	**Out**
Baseline	18.09	15.30	18.08	15.26
Word-based	18.10	7.71	25.62	12.67
Char-based	24.47	16.97	35.28	25.57
BPE-50	23.32	16.89	36.02	23.16
BPE-100	**25.12**	**18.7**	**38.16**	**26.38**
BPE-250	24.55	10.88	36.05	25.15
BPE-500	23.43	10.41	34.57	24.55
BPE-750	20.33	13.08	28.40	20.97
BPE-1000	21.58	12.56	24.76	18.36

5.1.2 Results base models for Dutch → Gronings

For the direction NL→ GRO, the in-domain baseline achieved a BLEU score of 18.09. We found a brevity penalty of 0.996, indicating that Groninger output sentences in general are shorter than Groninger reference sentences (Appendix A). Furthermore, the word-based model (18.10) was able to score slightly better than the baseline score (18.09). The character model achieved one of the best scores (24.27) along with the highest the N-gram precision (Appendix A). However, it received a brevity penalty for its short output (0.914). The BPE models yielded a relatively high score. The low number of merges model performed a bit better than the models with a high number of merges. BPE-100 scored the highest BLEU score (25.12). None of the BPE models received a brevity penalty.

For the out-domain, the baseline achieved a BLEU score of 15.3. This is, as expectable, lower than the baseline of the in-domain corpus. In addition, all scores are much lower compared to the in-domain scores. The character model seemed to be one of the best models (16.97) again. However, like the in-domain, it received a brevity penalty (0.917). Furthermore, the low merges BPE models scored the best of all models. The BPE-100 model (18.71) better than the BPE-50 (16.89). However, the N-gram precision (Appendix A) between the BPE-50 (51.9 on unigram level), BPE-100 (52.0) and the character model (51.6) are very close .

5.1.3 Results base models Gronings → Dutch

The BLEU scores in general in Table 13 for the direction GRO→ NL are relatively higher than for the direction NL→ GRO. Nevertheless, the same models produced the best scores for the direction GRO→ NL as we saw for the direction NL→ GRO. The character-based model (35.28 BLEU points) along with the BPE-50 (36.02) and BPE-100 (38.16) models produced the best score. Only the BPE-100 models did not receive a brevity penalty (Appendix A). In the out-domain test set, again, the same models performed the best (25.57 BLEU points for character-based, 23.16 and 26.38 BLEU points for BPE-50 and BPE-100 respectively). However, the gap of more than 10 BLEU points between the in- and out-domain might indicate overfitting since the models were trained on the in-domain test set but performed poorly on the out-domain test set.

Although the BPE-100 outperformed the character model, overall, the character model seemed to be quite robust in N-gram precision in comparison to the other models for both directions. The dominance of the character model is in line with the research of van Noord et al. (2018).

5.2 BACKTRANSLATION

As applied by Sennrich et al. (2015a), the method backtranslation was introduced in this thesis. We applied this technique with the Groninger monolingual books: Grimm, Biebel and Children Bible. The books were grouped and divided in subcorpora of 10,000, 20,000 and 40,000 sentences. For Dutch, we divided the Ted corpus in subcorpora of 10,000, 20.000, 40,000 and 80,000 sentences. The monolingual data was added to the GV training data, which provided us with a hybrid model in terms of data provenance (authentic data + synthetic data). As in the research of Poncelas et al. (2018), we added step wise more and more synthetic data to see if the BLEU score raised when evaluating.

5.2.1 The best base model for backtranslation

For backtranslation purposes, we opted for the character model since the gap between in- and out-domain was smaller and the N-gram precision was higher than the BPE models (Table 13 and Appendix A). Thereby, Ling et al. (2015) had already proven in their research that backtranslation was effective in combination with the character model. Therefore, the monolingual subcorpora were set to character en-

coding and added step wise to the training data. The amount of epochs trained for both directions are shown in Table 14.

Table 14: Epochs per language with backtranslation for character model

Model	NL→ GRO	GRO→ NL
character	57	42

5.2.2 Results backtranslation

The results of the backtranslation method are shown in Table 15. In contrast to the non-backtranslation results, all backtranslation models performed better, the in-domain slightly less than the out-domain. Nevertheless, the score gap between the in-domain and out-domain sets were becoming smaller compared to the non-backtranslation results. Although the scores were similar among the different steps. The 40k step showed the biggest improvement for both directions and domains (NL → GRO: 27.84(in), 26.20 (out), GRO → NL: 36.86 (in), 28.54 (out))

Table 15: BLEU scores with and without backtranslate on the character-model; **Bold** = best score obtained

Model	NL-GRO		GRO-NL	
	In	**Out**	**In**	**Out**
Char-based	24.47	16.97	35.28	25.5
Char-based + 10.000	27.01	24.92	35.37	27.19
Char-based + 20.000	27.43	24.07	34.79	25.18
Char-based + 40.000	**27.84**	**26.20**	**36.86**	**28.54**
Char-based + 80.000	NA	NA	34.00	23.85

5.2.3 Results iterative backtranslation

Due to the success of the backtranslation method, we applied the iterative back-translation as well. Therefore, we used our new best models, achieved with back-translation (Char-based + 40,000 model for both directions). After translating the 40,000 monolingual corpus once again, both models were re-trained and tested as in the research of Hoang et al. (2018). The amount of epochs trained per models are shown in Table 16. The results are shown in Table 17.

Table 16: Epochs per model with iterative backtranslation for character-model

Model	NL→ GRO	GRO→ NL
character	68	36

Table 17: BLEU scores with iterative backtranslation on the character-model; **Bold** = best score obtained

Model	NL-GRO		GRO-NL	
	In	Out	In	Out
non-iter	27.84	26.20	36.86	28.54
10.000	23.34	21.77	30.31	24.88
20.000	**29.01**	**27.99**	31.57	19.43
40.000	25.43	23.97	**37.75**	**28.25**
80.000	NA	NA	34.15	28.01

The simulation of the iterative backtranslation settings showed the best BLEU scores so far. For the direction NL→ GRO, the 20,000 model achieved a BLEU score of 29.01 for the in-domain and 27.99 for the out-domain. This was an improvement of 1.17 BLEU points for the in-domain and 1.21 BLEU points for the out-domain compared to non-iterative backtranslation. With these results, the gap between in and out-domain became again smaller, which may indicate less overfitting. For the GRO→ NL direction our 40,000 monolingual model show an improvement in BLEU score for the in-domain set (37.75 compared to 36.86) and a little decline in BLEU score for the out-domain set (28.25 compared to 28.54).

5.2.4 Best models so far after iterative backtranslation

The best models are shown in Table 18. These are the 20,000 sentences with iterative backtranslation model for the direction NL→ GRO and the 40,000 sentences with iterative backtranslation model for the direction GRO→ NL.

Table 18: Best models so far for both directions measured in BLEU score

Direction	Corpus	Method	BLEU In	BLEU Out
NL-GRO	20,000 mono	iterative backtranslation	29.01	27.99
GRO-NL	40,000 mono	iterative backtranslation	37.75	28.25

Although the gap between in and out- domain decreases since the base model, there was still a gap which suggested a form of overfitting. Therefore, we applied a BPE approach to find out if we could see any difference in score. Table 13 in Section

5.1 shows that the BPE-100 model outperformed the other BPE models. We tested if the BPE model with 100 merges performed different in the same settings in contrast to the character model.

In order to do this, the subcorpora (20.000 Gronings monolingual and 40.000 Dutch monolingual) were set to BPE-100 encoding and added to the training set. The amount of epochs trained for the BPE-100 models are shown in Table 19. The results of the BPE model are shown in Table 20 compared to the character model.

Table 19: Epochs per model with iterative backtranslation for the BPE-100 model

Model	NL→ GRO	GRO→ NL
BPE-100	91	77

For the direction GRO → NL a small decrease in BLEU points is shown for the BPE-100 model (35.03 in-domain and 26.33 out-domain) compared to the character model (37.75 in-domain and 28.25 out-domain). The BPE model for the NL→ GRO direction performed better (31.18 in-domain and 29.63 out-domain) compared to the character model (29.01 in-domain and 27.99 out-domain).

Table 20: Best models in comparison for character and BPE in same settings ; **Bold** = best score per direction

Direction	Model	Corpus	Method	BLEU In	BLEU Out
NL-GRO	Char	20,000 mono	iterative backtranslation	29.01	27.99
NL-GRO	BPE-100	20,000 mono	iterative backtranslation	**31.18**	**29.63**
GRO-NL	Char	40,000 mono	iterative backtranslation	**37.75**	**28.25**
GRO-NL	BPE-100	40,000 mono	iterative backtranslation	35.03	26.33

Since the BLEU scores of both models were close, we applied the ensembling method to the character model as well to the BPE-100 model.

5.3 ENSEMBLE

The models described in Table 20 were used to see if we could enhance the BLEU scores. We applied the ensemble method used in the research of Luong et al. (2015) on our best models so far. We took the best three, six and eight models and evaluated them on both of our test corpora. Table 21 shows the results of the ensemble method with character- and BPE-100 encoding.

Table 21: Mulit-BLEU scores for character encoding in combination with ensemble; **Bold** = best score obtained

Models	**CHAR**				**BPE-100**			
	NL-GRO		**GRO-NL**		**NL-GRO**		**GRO-NL**	
	In	**Out**	**In**	**Out**	**In**	**Out**	**In**	**Out**
No ensemble	29.01	**27.99**	37.75	28.25	**31.18**	29.63	35.03	26.33
3 models	28.64	26.41	37.73	**29.68**	31.07	29.92	37.61	27.99
6 models	29.06	25.97	**38.14**	29.53	30.60	30.19	38.69	29.03
8 models	**29.21**	26.05	37.89	29.65	30.57	**30.42**	**38.84**	**29.77**

For the character model (NL→ GRO) all scores of the models are around 29 BLEU points for the in-domain test set and around 26 BLEU points for the out-domain set. They do not differ much from the model without ensembling. In the in-domain column, the eight (29.21 in-domain and 26.05 out-domain) and six (29.06 in-domain and 25.97 out-domain) models perform slightly better. For the direction GRO→ NL, the scores are (almost) all higher than without ensembling. The six model ensemble achieved with a BLEU score of 38.14 for the in-domain and 29.53 for the out-domain set the best score. This is an improvement of one BLEU point for the out-domain set which corresponds with the study of Luong et al. (2015).

For the BPE-100 (NL→ GRO), the ensemble method did not surpass the previous score for the in-domain (31.18 BLEU points). However, for the eight models ensembling on the out-domain set there was a small improvement (30.42 BLEU points) compared to the non-ensemble score. Nevertheless, the three models perform relatively the best on in- and out-domain combined (31.07 in-domain and 29.92 out-domain). The direction GRO→ NL had a bigger improvement of 3.81 BLEU points for the in-domain set (38.84) and an improvement of 3.44 BLEU points for the out-domain set (29.77). However, the gap between in- and out-domain for the GRO→ NL remained big (9.05 BLEU points), but compared to the character encoding model it is a small improvement. Only the in-domain NL→ GRO direction received a brevity penalty for all ensemble models (Appendix A). For all other models, the output sentences were as long or longer as the reference sentence.

5.4 OVERVIEW AND ANALYSIS

An overview of our best modes per direction is given in Table 22 and Table 23 as well with the improvements calculated with respect to the baseline.

Table 22: Best models and improvement for the direction NL→ GRO; Backtrans= backtranslation, IB = iterative backtranslation, syn = synthetic data, number = amount of ensemble models

Corpus	Encoding	Method	In	Out	Improv in	Improv out
NA	Word	NA	18.08	15.30	Baseline	Baseline
GV	Char	NA	24.47	16.97	+6.39	+1,67
GV + 40k syn	Char	Backtrans	27.84	26.20	+9.76	+10.90
GV + 20k syn	Char	IB	29.01	27.99	+10.93	+12.69
GV + 20k syn	BPE	IB + three	31.07	29.92	+12.99	+14.62

Table 23: Best models and improvement for the direction GRO→ NL ; Backtrans= backtranslation, IB = iterative backtranslation, syn = synthetic data, number = amount of ensemble models

Corpus	Encoding	Method	In	Out	Improv in	Improv out
NA	Word	NA	18.08	15.26	Baseline	Baseline
GV	Char	NA	35.28	25.57	+17.20	+10.31
GV+ 20k syn	Char	Backtrans	36.86	28.54	+18.78	+10.28
GV+ 40k syn	Char	IB	37.75	28.25	+19.67	+12,99
GV+ 40k syn	Char	IB + six	38.14	29.53	+20.06	+14.27
GV+ 40k syn	BPE	IB + eight	38.84	29.77	+20.76	+14.51

These Tables reflect the effectiveness of the translation models. For the direction NL→ GRO the final model managed to reach a BLEU score of 31.07 (in) and 29.92 (out), an improvement of +12.99 and +14.62 points above baseline. For the direction GRO→ NL our best model managed to reach a BLEU score of 38.84 (in) and 29.77 (out). This is an improvement of +20.76 and +14.51 points compared to the baseline score.

5.4.1 Sentence Bucket Analysis

In this section a more advanced comparison of the output is given by our best models. This was done with the Python 3 library Compare MT (Neubig et al., 2019). This package allowed us to do a Sentence Bucket analysis by various statistics (e.g. sentence BLEU, length differences with the reference and overall length). In Table 24 the BLEU score per sentence length is given for our best character and BPE-100 models.

Table 24: Overview in-depth sentence length of character model in BLEU score; **Bold** = best score obtained

Sentence length	CHAR				BPE-100			
	NL-GRO		GRO-NL		NL-GRO		GRO-NL	
	In	Out	In	Out	In	Out	In	Out
<10	31.00	**29.9**	**42.86**	31.72	34.19	31.54	40.75	29.03
(10,20)	**32.91**	27.92	42.81	30.15	**35.95**	28.66	41.60	**31.62**
(20,30)	26.95	29.08	33.09	27.46	29.08	**32.00**	38.67	30.30
(30,40)	25,02	25.09	33.33	**35.73**	24.75	27.40	34.26	28.35
(40,50)	17,24	19.46	7.62	7.97	20.62	29.62	27.32	28.3
(50,60)	12.89	22.79	10.49	10.46	35.12	10.22	**42.72**	13.40
>=60	29.89	0.00	11.35	0.00	11.12	0.00	6.99	6.11

As shown in Table 24 the character model performed quite reasonable in short sentences compared to the longer ones. The BPE, like the character model, performed better in short sentences. Nevertheless, the BPE model seemed to do a better job when translating the longer sentences. Still, both models seemed to struggle to translate sentences longer than 60 words in the out-domain test set. The poor performances in long sentences might suggest that there were no long sentences present in the training data. Furthermore, the BLEU scores for the out-domain are higher with the middle long sentences than for the in-domain (e.g. for sentence length 20-30 for character in-domain: 26.95 out-domain: 29.08, for BPE-100 in-domain: 29.08 out-domain: 32.00).

5.4.2 Analysis of translations

Since BLEU scores do not tell everything about the correctness of the translation, we took a closer look at the translations produced by our best models. Because the scores of our models are close, we looked at the character model and BPE model to see any differences. Random sentences of the in- and out-domain were retrieved from the test data and are shown in Table 25.

Table 25: In depth review of the analysis for the direction NL→ GRO

Form	Domain	Example sentences
Source	IN	ik ben blij dat u er bent , een mens wil ook wel een echt gesprek .
Reference		ik bin blied dat ie der binnen , men wil ook wel ais mit ain oetproaten .
Hypothesis Char		ik bin blied dat ie der binnen , n mensk wil ook wel n echt gesprek .
Hypothesis BPE		ik bin blied dat ie der binnen , n mensk wil ook wel n echt gesprek .
Source	OUT	hij praatte tegen ons alsof we even oud waren .
Reference		hai pruit tegen ons of wazzen wie glieke old .
Hypothesis Char		hai proatte tegen ons as asof we even ol woaren .
Hypothesis BPE		hai proatte tegen ons of we even oven old wazzen .

In Table 25 in the in-domain source sentence we see the pronouns "ik" and "u". As we described the pro-drop as one of the characteristics of Gronings (see Section 3.1.1), we see that the Groninger reference sentences, as expectable, dropped the pronoun "u" and replaced it with the word "ien". In the hypothesis sentence, both of our systems performed the same pro-drop as in the reference sentence. However, in the out-domain sentence, the comprehensive word "wazzen" covers the pronoun "we" and the conjunction word "alsof". The system did not drop the pronoun and kept the word "we".

Another distinctive of Gronings is word order. The order of the hypothesis sentences in the in-domain corresponds completely with the reference sentences before the comma. However, after the comma, the word order changed as the order corresponded more toward the Dutch source sentence. A mix between Dutch and Gronings became more visible in the out-domain sentence.

In the out-domain, the BPE model had the word "wazzen" in its vocabulary as the character model did not. However, the word order is not correct as the word "wazzen" was put at the end of the sentence. Furthermore, the BPE model added an additional unnecessary word "oven" in its final output.

Table 26: In depth review of the analysis for the direction GRO→ NL

Form	Domain	Example sentences
Source		bruier , joag dat mensk vot zeg 'k die !
Reference	IN	broer , jaag dat mens weg zeg ik je !
Hypothesis Char		broer , jaag dat mens voet zeg ik je !
Hypothesis BPE		broer , jaag dat mensen vot zeg ik je !
Source		woarom kikst mie nooit meer aan .
Reference	OUT	waarom kijk je me niet meer aan ?
Hypothesis Char		waarom kijkst me nooit meer aan .
Hypothesis BPE		waarom kikst me nooit meer aan .

For the direction GRO→ NL the same issues occurred in reversed order as seen in Table 26. In the in-domain sentence for the character model the word "voet" was incorrect. The system seemed to have issues replacing irregular verbs of Gronings with verbs of Dutch. In the out-domain sentences the pro-drop was still visible in the hypothesis sentence. Where the verb "kijkst" should have been replaced with the verb "kijk" and pronoun "je", the system found a combination between the Groninger word "kikst" and the Dutch word "kijk" and ended up with "kijkst". Thereby, leaving out the pronoun "je" completely.

The BPE output showed two differences in the in-domain sentence. First the word "mensen" was plural instead of "mens". In comparison, the character model did it correctly. Second, the BPE system kept the word "vot" instead of the Dutch "weg". In the out-domain two more differences occurred. The BPE translation chose for the word "kikst" instead of "kijk". Furthermore, just like the character translation the Dutch pronoun "je" was left out.

The sentence retrieved in Table 26 are a small example of the translation given by the systems. More translations with BLEU score per sentence can be found in Appendix B. These comparisons were created with the Compare MT function which were described in Section 5.4.1. The system provided several sentences where one system performed better than the other (in our case character versus BPE-100).

5.5 WEB API

To implement our best models in a functional environment, a web API was created with Flask[1]. Flask is a lightweight Python based web application framework. It is

[1] http://flask.pocoo.org

designed to getting started quick and easy, with the ability to scale up to complex applications.

To run our models locally in a Flask web environment, we were required to train our systems again but this time on CPU. The models were trained on 24 nodes, on a 28 cores @ 2.4 GHz, two Intel Xeon E5 2680v4 CPU's. We used the same settings as described in Table 12 only for the USE_CUDNN parameter which was set to false (since this parameter is for GPU's only). By training on CPU the training time became increasingly longer. An overview of the amount of epochs trained is given in Table 27.

Table 27: Epochs per model on CPU for BPE-100 and character encoding model per language

Model	NL→ GRO	GRO→ NL
BPE -100	47	50
character	23	27

Flask is a minimalist (or micro) framework which refrains from imposing the way critical things are handled. Instead, Flask allows the developers to use the tools they desire and are familiar with. For this purpose, it comes with its own extensions index and a good amount of tools already exist to handle pretty much everything. In addition, Flask has its own architecture and therefore the developer can create its own workflow relatively easy. A flowchart of our NMT system in combination with Flask is given in Figure 4.

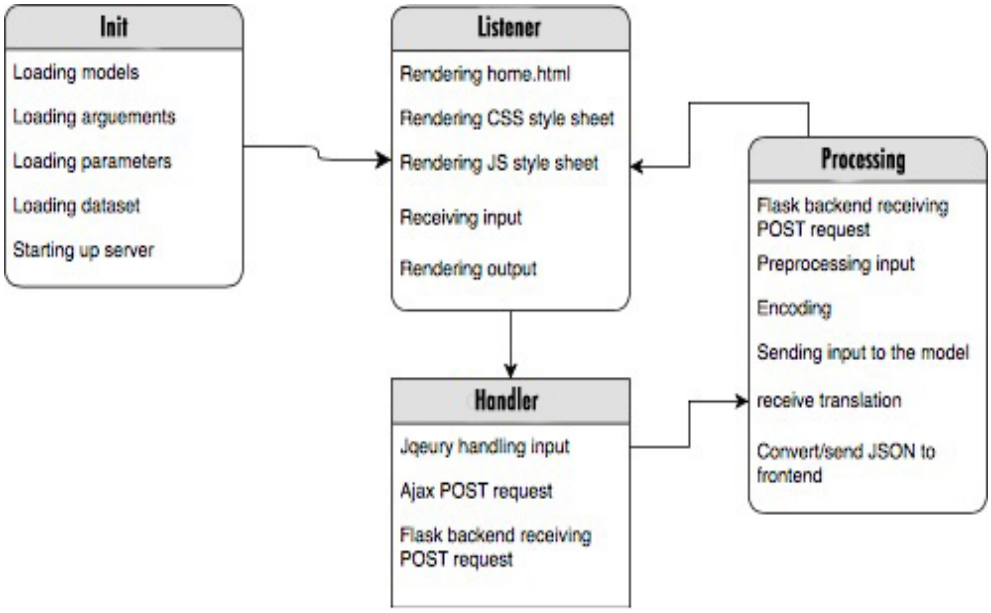

Figure 4: Flowchart for the Flask web API

When executing the Flask environment, the API starts up by loading all the requirements before starting up the server. This includes the models (BPE and

character), the corresponding arguments, parameters and datasets (init component). Next, the Flask server will boot up and the server starts running locally. When running, the first task is rendering the HTML file. Since the site is a single-page only website, only the homepage has to be rendered. The homepage was constructed with the help of Bootstrap[2]. Next, it will render the corresponding CSS and JQuery files. On the homepage, the user can set the direction to NL→ GRO or to GRO→ NL, in addition, the user can set the encoding to BPE or character encoding as seen in Figure 5.

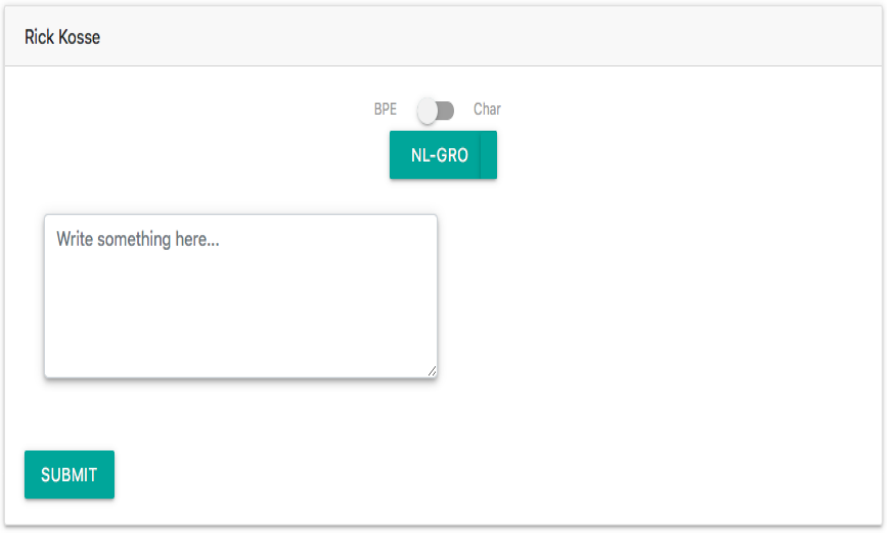

Figure 5: Example of homepage web API

By clicking on the submit button the user can send its input to the back-end (listener component). The JQuery script takes the input and via an AJAX POST request, it hands the input over to the Flask back-end. When the POST method is validated, Flask calls several Python scripts in order to preprocess, encode, translate, decode and restore. The Flask API returns a JSON file to the JQuery script (processing component). When validated successful, the HTML frontpage outputs the translation next to the input the user has given as seen in Figure 6. When the user desires, the user can give a new input by typing a new sentence and clicking on the submit button again.

[2] https://getbootstrap.com

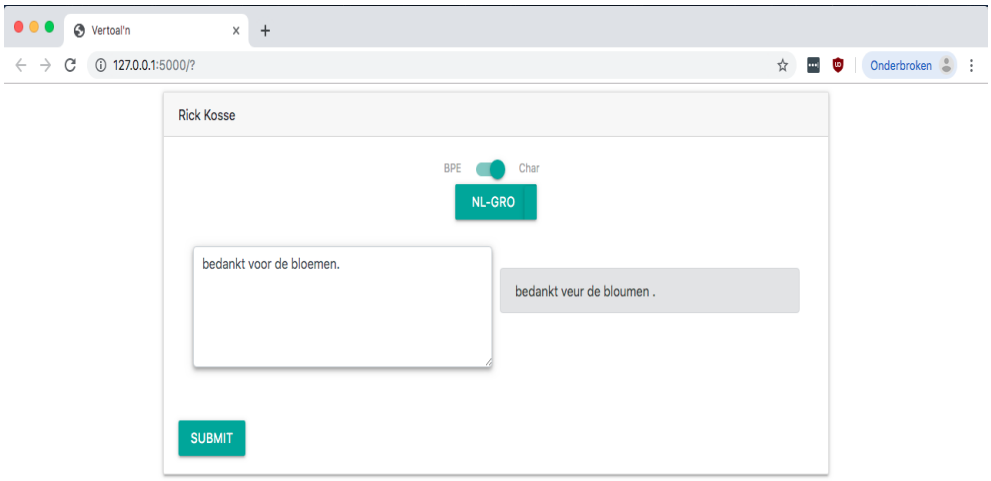

Figure 6: Example of homepage web API and its output

So far the web API runs locally, in order to the deploy the site, de-preprocessing needs to be implemented. Currently when a user inputs a sentence which for example contains capital letters, all these capital letters are lowercased by the preprocessing script. However, after translating, the capital letters are not set back. Furthermore, the model is completely trained on sentences with punctuation at the end. When an input without any punctuation is given, the model does not translate correctly. An example of a translation without punctuation is given in Figure 7.

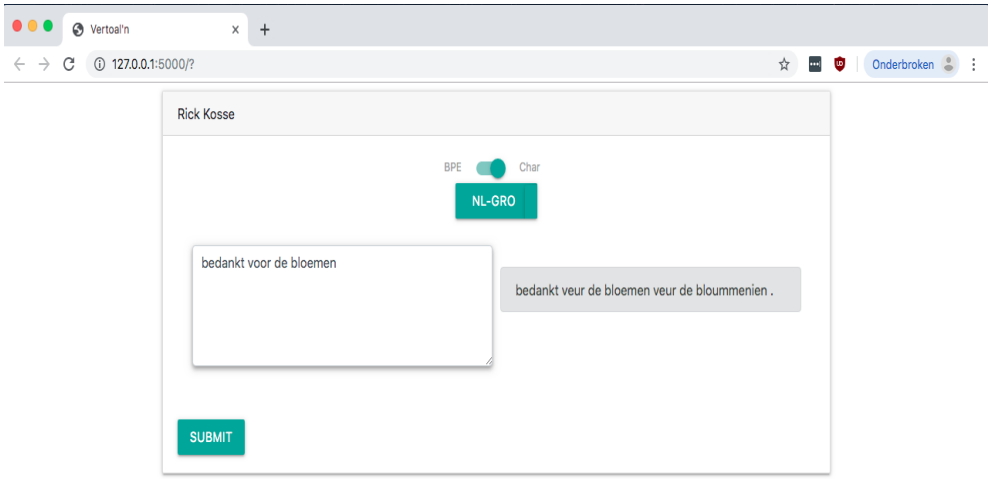

Figure 7: Example of homepage web API and its output without punctuation

Therefore, a punctuation- and capital letter script has to be created and implemented in the Flask back-end. With the punctuation script the back-end can put a punctuation behind the user input when it is not present in order to translate more correctly.

5.6 DISCUSSION

With the analysis of the translation we can conclude that we made reasonable translations, since the sentences largely correspond to the reference sentences as seen in Section 5.4.2. However, the system seems to have difficulties with Groninger words that do not originate from Dutch. In Table 25 the words "glieke old" do not originate from Dutch but from German (gleich alt). We see that the program picks for different, more Dutch related solutions. Hereby we conclude that we need more data to get accurate translations. In general we would not call our translations flawless Gronings, since our system output sentences consist of a mix of Dutch and Gronings. In order to retrieve more data and thereby tackle the problem of regional differences we recommend, for further research, to use hand annotated data instead. The data can be created by translating Dutch to Gronings by people from different regions. This way the data can easily be divided by region. Eventually, the system can translate several regional dialects.

Furthermore, we struggled with the evaluation of the translations. The book Martha, which contained Groninger sentences, might contain a different variety of Gronings than the system was trained on, which could have influenced the translation results. Due to the scarcity of data there were no other options. Again, hand annotated data could be an option for evaluating purposes. Also related to the evaluation, in this research we focused mainly on the outcome of BLEU scores. However, BLEU does not fully capture the quality of a translation. In the field of MT more evaluation metrics are known like NIST (Lin and Och, 2004) which also calculates how informative a particular n-gram is or METEOR (Lavie and Agarwal, 2007) and Word Error Rate (Klakow and Peters, 2002). Furthermore, statistical tests were required in order to prove that one NMT system is significantly better than the other one (like character versus BPE-100). Regarding the training process, cross validation was needed to see if the systems were not overfitting. In this research we saw a gap between the in-domain and out-domain test set, which might indicate overfitting. Due to a combination of time constraints and the large amount of training time of the systems, we were not able to achieve this within this thesis.

On the NMT side, for further research, we used a standard NMT system and focused primarily on preprocessing. Testing different systems or performing a grid search might result in higher scores. Finally, in this research, two systems in two directions were tested. One might say that these systems are worth two independent researches rather than one. In this research we did not have the time to focus on the individual behaviour of the two systems.

6 CONCLUSION

In this thesis we aimed to build a machine translation system for Dutch to Gronings and vice versa, despite the scarcity of resources. Our attentional seq2seq system and ensemble method showed that with iterative backtranslation in combination with hybrid data, reasonable translations were possible. However, we have to take into account that Gronings is closely-related to Dutch as shown with our baseline BLEU score of 15. Nevertheless, we showed that with our best model we were able to surpass the baseline score to a BLEU score of 31.07 (in-domain) and a BLEU score of 29.92 (out-domain). This was an improvement of +12.99 and +14.62 with respect to our baseline. For our GRO→ NL model we even achieved better results with a BLEU score of 38.84 (in-domain) and 29.77 (out-domain), an improvement of +20.76 and +14.51. Our final model translates reasonably although it must be said that the translations are often too much of a mix between Dutch and Gronings. For further research we recommend more hand annotated training data.

This research builds on the field of machine translation. With this thesis and its system we hope to contribute a little to prevent the end of Gronings.

BIBLIOGRAPHY

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Álvaro Peris and F. Casacuberta (2018). NMT-Keras: a Very Flexible Toolkit with a Focus on Interactive NMT and Online Learning. *The Prague Bulletin of Mathematical Linguistics 111*, 113–124.

Bahdanau, D., K. Cho, and Y. Bengio (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Berendsen (2019). Gronings — wikipedia, de vrije encyclopedie. [Online; accessed 2-juli-2019].

Bergstra, J., O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio (2010). Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, Volume 4. Austin, TX.

Binsbergen e.a, L. v. (2008). *Biebel in t Grunnegers*. NBG/Jongbloed - Heerenveen.

Blunsom, P. and E. Grefenstette (2013). Nal kalchbrenner, et al. 2014. a convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chollet, F. et al. (2015). Keras. https://keras.io.

Costa-Jussa, M. R. and J. A. Fonollosa (2016). Character-based neural machine translation. *arXiv preprint arXiv:1603.00810*.

Dijken, M. v. (2016). *Grunneger Grimm*. Stichting t Grunneger Bouk.

Forcada, M. L., M. Ginestí-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, J. A. Pérez-Ortiz, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F. M. Tyers (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine translation 25*(2), 127–144.

Gale, W. A. and K. W. Church (1993). A program for aligning sentences in bilingual corpora. *Computational linguistics 19*(1), 75–102.

Gompel, M. v., A. van den Bosch, A. Dijkstra, and A. Dijkstra (2014). Oersetter: Frisian-dutch statistical machine translation.

Hoang, V. C. D., P. Koehn, G. Haffari, and T. Cohn (2018). Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 18–24.

Kim, Y., Y. Jernite, D. Sontag, and A. M. Rush (2016). Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Klakow, D. and J. Peters (2002). Testing the correlation of word error rate and perplexity. *Speech Communication 38*(1-2), 19–28.

Lavie, A. and A. Agarwal (2007). Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 228–231. Association for Computational Linguistics.

Lin, C.-Y. and F. J. Och (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pp. 605. Association for Computational Linguistics.

Ling, W., I. Trancoso, C. Dyer, and A. W. Black (2015). Character-based neural machine translation. *arXiv preprint arXiv:1511.04586*.

Liudgerstichten (2007). *Biebel in t Grunnegers*. Jongbloed.

Luong, M.-T., H. Pham, and C. D. Manning (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Moore, R. C. (2002). Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pp. 135–144. Springer.

Neco, R. P. and M. L. Forcada (1997). Asynchronous translations with recurrent neural nets. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, Volume 4, pp. 2535–2540. IEEE.

Neubig, G., Z.-Y. Dou, J. Hu, P. Michel, D. Pruthi, and X. Wang (2019, June). compare-mt: A tool for holistic comparison of language generation systems. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL) Demo Track*, Minneapolis, USA.

Otte, P., P. Mulierlaan, and F. M. Tyers (2011). Rapid rule-based machine translation between dutch and afrikaans. In *Proceedings of the 15th conference of the European Association for Machine Translation, 30-31 May 2011, Leuven, Belgium*, pp. 153–160.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics.

Pecina, P., A. Toral, A. Way, V. Papavassiliou, P. Prokopidis, and M. Giagkou (2011). Towards using web-crawled data for domain adaptation in statistical machine translation.

Pettersson, E., B. Megyesi, and J. Tiedemann (2013). An smt approach to automatic annotation of historical text. In *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013; May 22-24; 2013; Oslo; Norway. NEALT Proceedings Series 18*, Number 087, pp. 54–69. Linköping University Electronic Press.

Poncelas, A., D. Shterionov, A. Way, G. M. de Buy Wenniger, and P. Passban (2018). Investigating backtranslation in neural machine translation. *CoRR abs/1804.06189*.

Post, M., C. Callison-Burch, and M. Osborne (2012). Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pp. 401–409. Association for Computational Linguistics.

Reker, S. (2008). *Goud Volk*. in Boekvorm Uitgevers bv.

Sennrich, R., B. Haddow, and A. Birch (2015a). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Sennrich, R., B. Haddow, and A. Birch (2015b). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Sutskever, I., O. Vinyals, and Q. V. Le (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112.

Toral, A., M. Poch, P. Pecina, and G. Thurmair (2012). Efficiency-based evaluation of aligners for industrial applications. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation: EAMT 2012; 2012 May 28-30; Trento, Italy. Trento: Fondazione Bruno Kessler; 2012. p. 57-60.* European Association for Machine Translation.

Unhammer, K. and T. Trosterud (2009). Reuse of free resources in machine translation between nynorsk and bokmål.

van Noord, R., L. Abzianidze, A. Toral, and J. Bos (2018). Exploring neural methods for parsing discourse representation structures. *arXiv preprint arXiv:1810.12579*.

Varga, D., P. Halácsy, A. Kornai, V. Nagy, L. Németh, and V. Trón (2007). Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4 292*, 247.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.

Visscher, K. (2004). *MARTHA*. Noordboek.

Wikipedia-contributors (2019). Gronings — wikipedia, de vrije encyclopedie. [Online; accessed 2-juli-2019].

Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

A N-GRAM PRECISION AND BREVITY PENALTY

Table 28: Multi-Bleu and N-grams precision scores Dutch to Gronings In-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
Baseline	18.09	49.2	22.9	12.3	7.8	0.996
Word-based	18.10	48.2	24.1	12.7	7.3	1.000
Char-based	24.47	58.5	33.4	20.4	12.9	0.914
BPE-50	23.32	51.8	29.8	17.7	10.8	1.000
BPE-100	25.12	54.6	31.7	19.3	12.0	1.000
BPE-250	24.55	55.4	31.4	18.5	11.3	1.000
BPE-500	23.43	52.2	30.1	17.9	10.7	1.000
BPE-750	20.33	47.1	26.5	15.2	9.0	1.000
BPE-1000	21.58	47.1	27.6	16.4	10.2	1.000

Table 29: Multi-Bleu scores Dutch to Gronings Out-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
Baseline	15.30	49.6	21.5	10.3	5.4	0.979
Word-based	7.71	34.3	11.4	4.6	2.0	1.000
Char-based	16.97	51.6	25.2	13.2	6.8	0.917
BPE-50	16.89	51.9	26.8	14.1	7.6	0.861
BPE-100	18.71	52.0	25.5	13.4	7.0	0.995
BPE-250	10.88	43.0	17.0	7.1	3.0	0.973
BPE-500	10.41	39.5	15.0	6.6	3.0	1.000
BPE-750	13.08	44.5	19.7	9.7	4.8	0.920
BPE-1000	12.56	39.3	17.1	8.5	4.4	0.998

Table 30: Multi-Bleu scores Gronings To Dutch In-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
Baseline	18.08	49.0	22.8	12.2	7.8	1.000
Word-based	25.62	55.0	31.6	19.7	12.6	1.000
Char-based	35.28	65.6	42.8	29.4	20.7	0.976
BPE-50	36.02	67.4	44.1	30.2	21.1	0.972
BPE-100	38.16	68.2	44.9	31.2	22.2	1.000
BPE-250	36.05	67.7	44.2	30.4	21.7	0.962
BPE-500	34.57	65.0	41.9	28.4	19.9	0.981
BPE-750	28.40	56.9	35.7	23.5	15.6	0.967
BPE-1000	24.76	46.1	30.0	20.1	13.5	1.000

Table 31: Multi-Bleu scores Gronings To Dutch Out-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
Baseline	15.26	48.6	21.0	10.1	5.3	1.000
Word-based	12.67	41.6	17.5	8.3	4.2	1.000
Char-based	25.57	57.7	32.6	19.5	12.1	0.991
BPE-50	23.16	58.9	32.9	19.9	12.4	0.882
BPE-100	26.38	58.6	32.9	20.1	12.5	1.000
BPE-250	25.15	57.8	31.9	18.8	11.5	1.000
BPE-500	24.55	56.3	31.4	18.5	11.2	0.998
BPE-750	20.97	49.5	26.4	15.5	9.6	1.000
BPE-1000	18.36	45.8	23.7	13.4	7.8	1.000

Table 32: Backtranslate Char-model Multi-Bleu scores Dutch To Gronings In-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
10.000	27.01	59.2	34.8	21.6	14.1	0.960
20.000	27.43	59.2	34.8	21.4	13.8	0.983
40.000	27.84	58.9	34.5	21.4	13.8	1.000

Table 33: Backtranslate out-domain Char-model Multi-Bleu scores Dutch To Gronings Out-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
10.000	24.92	58.8	32.7	19.5	12.0	0.961
20.000	24.07	57.8	32.0	18.8	11.0	0.969
40.000	26.20	58.0	32.9	20.1	12.3	1.000

Table 34: Backtranslate Char-model Multi-Bleu scores Gronings To Dutch In-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
10.000	35.37	65.7	42.5	28.9	20.1	0.991
20.000	34.79	64.7	41.9	28.7	20.2	0.982
40.000	35.42	64.6	42.1	28.6	20.2	1.000
80.000	34.00	62.3	40.4	27.4	19.4	1.000

Table 35: Backtranslate out-domain Char-model Multi-Bleu scores Gronings To Dutch Out-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
10.000	27.19	58.6	34.0	20.8	13.2	1.000
20.000	25.18	57.2	32.2	19.3	11.9	0.986
40.000	26.70	58.0	33.5	20.4	12.8	1.000
80.000	23.85	53.7	30.0	17.9	11.2	1.000

Table 36: Re-backtranslate Char-model Multi-Bleu scores Gronings To Dutch In-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
10.000	30.31	59.8	37.1	23.9	15.9	1.000
20.000	31.57	61.8	37.6	24.9	17.2	1.000
40.000	37.75	66.4	44.4	30.9	22.3	1.000
80.000	34.15	62.9	40.6	27.5	19.4	1.000

Table 37: Re-backtranslate out-domain Char-model Multi-Bleu scores Gronings To Dutch Out-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
10.000	24.88	55.4	31.2	18.8	11.8	1.000
20.000	19.43	50.2	24.8	13.9	8.3	1.000
40.000	28.25	59.3	35.1	21.9	13.9	1.000
80.000	28.01	58.8	34.7	21.8	13.8	1.000

Table 38: Re-backtranslate Char-model Multi-Bleu scores Dutch to Gronings In-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
10.000	23.34	58.5	32.4	18.7	11.4	0.925
20.000	29.01	61.4	36.2	22.5	14.9	0.987
40.000	25.43	58.0	32.4	19.1	11.9	0.996

Table 39: Re-backtranslate out-domain Char-model Multi-Bleu scores Dutch To Gronings Out-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
10.000	21.77	58.4	31.1	17.8	10.4	0.904
20.000	27.99	61.7	35.5	21.8	14.1	0.977
40.000	23.97	59.1	31.7	18.2	10.9	0.971

Table 40: Embbedings in-depth

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
GV NL→ GRO	26.39	58.2	33.2	19.9	12.6	1.000
Martha NL→ GRO	24.32	56.7	30.6	18.1	11.1	1.000
GV GRO→ NL	36.82	65.4	43.2	30.1	21.6	1.000
Martha GRO→ NL	26.61	57.3	33.2	20.5	12.8	1.000

Table 41: Ensemble Gronings To Dutch in-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
3	37.73	66.6	44.2	30.9	22.3	1.000
6	38.14	67.1	44.8	31.3	22.5	1.000
8	37.8 9	66.9	44.5	31.0	22.3	1.000

Table 42: Ensemble Gronings To Dutch out-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
3	29.68	60.2	36.2	23.3	15.3	1.000
6	29.53	60.2	36.2	23.1	15.1	1.000
8	29.65	60.3	36.3	23.3	15.2	1.000

Table 43: Ensemble Dutch To Gronings in-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
3	28.64	61.7	36.5	22.6	14.7	0.974
6	29.06	62.1	36.7	22.6	14.8	0.983
8	29.21	62.2	36.9	22.9	14.9	0.982

Table 44: Ensemble Dutch To Gronings out-domain

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
3	26.41	61.3	34.3	20.8	12.9	0.963
6	25.97	61.4	34.5	21.1	13.2	0.937
8	26.05	61.6	34.5	21.1	13.1	0.941

Table 45: Ensemble Dutch To Gronings in-domain BPE

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
No ensemble	31.07	63.4	38.7	24.9	16.8	0.977
3	31.07	63.4	38.7	24.9	16.8	0.977
6	30.60	62.9	38.3	24.5	16.4	0.976
8	30.57	62.9	38.2	24.4	16.3	0.979

Table 46: Ensemble Dutch To Gronings out-domain BPE

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
No ensemble	29.92	62.7	36.7	23.2	15.0	1.000
3	29.92	62.7	36.7	23.2	15.0	1.000
6	30.19	62.9	36.9	23.4	15.3	1.000
8	30.42	63.2	37.1	23.7	15.4	1.000

Table 47: Ensemble Gronings To Dutch in-domain BPE

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
No ensemble	35.03	64.8	41.9	28.3	19.6	1.000
3	37.61	67.0	44.4	30.7	21.9	1.000
6	38.69	68.2	45.4	31.7	22.8	1.000
8	38.84	68.4	45.8	31.9	22.8	1.000

Table 48: Ensemble Gronings To Dutch out-domain BPE

Corpus	Bleu	1-gram	2-gram	3-gram	4-gram	BP
No ensemble	26.33	57.9	32.8	19.9	12.7	1.000
3	27.99	59.8	34.6	21.5	13.8	1.000
6	29.03	60.4	35.4	22.5	14.7	1.000
8	29.77	61.2	36.2	23.2	15.3	1.000

B SENTENCE ANALYSIS

Table 49: Sentences where the BPE encoding outperforms the Character encoding at sentence-level for in-domain NL→ GRO

Form	Example sentences	BLEU score
Ref:	vuil laiver hou ik mie vuil .	
BPE:	vuil laiver hou ik mie vuil .	100.
Char:	voel laiver hou ik mie voel .	54.1082
Ref:	nee , nee wicht , kiek , dat main ik nou nait meer !	
BPE:	nee , nee wicht , kiek , dat vind ik nou nait meer !	80.0320
Char:	nee , nee maid , kiek , dat vin ik nuit meer !.	37.0533
Ref:	(staait ien deur .) .	
BPE:	(staait ien deur .) .	100
Char:	(staait in deur .)	48.8923

Table 50: Sentences where the Character encoding outperforms the BPE encoding at sentence-level for in-domain NL→ GRO

Form	Example sentences	BLEU score
Ref:	hol die stil over dien mientje .	
BPE:	hol dien stil over die mientje .	34.5721
Char:	hol die stil over dien mientje .	100
Ref:	dat is dien verdainde loon !	
BPE:	dat is die verdainde loon !	48.8923
Char:	dat is dien verdainde loon !	100
Ref:	heur ie dat moeke ?	
BPE:	heurt joe dat moeke ? ’ .	41.1134
Char:	hoort ie dat moeke ?	75.9836
Ref:	dut ’n stap achteroet .)	
BPE:	dut n stap achteruit .)	33.0316
Char:	dut n stap achteroet .)	64.3459

Table 51: Sentences where the BPE encoding outperforms the Character encoding at sentence-level for the out-domain NL→ GRO

Form	Example sentences	BLEU score
Ref:	noast aaltje , doar zat martha aaltied .	
BPE:	noast allie , doar zat martha aaltied .	75.0624
Char:	noast aalie , doar zat marthoa altied .	33.0316
Ref:	n goederntraain schoof bedoard veurbie en veur t eerst vergat ik om de woagens te tellen .	
BPE:	n goederentrein schoof bedoard veurbie en veur t eerst vergat ik de woagons te tellen .	61.3572
Char:	n gouderntrain schoof bedoard voorbie en veurbie en vergat ik de wagons te tellen .	20.9708
Ref:	mor aal voaker was t roak .	
BPE:	mor steeds voaker was t roak.	70.7107
Char:	mor steeds voaker was ’t roak .	32.1729
Ref:	denkst doe wel ais aan doodgoan ? ”	
BPE:	denk doe wel ais aan doodgoan ? ’	72.5980
Char:	denk doe wel ains aan doodgoan ?	35.6403

Table 52: Sentences where the Character encoding outperforms the BPE encoding at sentence-level for the out-domain NL→ GRO

Form	Example sentences	BLEU score
Ref:	over t smale padje noast de raals kwammen de duutsers , bedoard mor woaksoam .	
BPE:	over t smalle poadje noast de rails kwammen de doetsers , bedoard man woakzoam .	18.3938
Char:	over 't smale poadje noast de raais kwammen de duutsers , bedoard mor woakzoam .	44.2850
Ref:	op dit stee huif je nait noar spouken te zuiken .	
BPE:	op dizze plek huifke nait noar spoken te zuiken .	26.2383
Char:	op dizze plek houfke nait noar spouken te zuiken .	53.3896
Ref:	klik , klik , klik , zee de wekker in de stilte .	
BPE:	klik , klik , klik proatte de wekker ien stilte .	42.9935
Char:	klik , klik , klik , klik proatte de wekker in de stilte .	72.4158
Ref:	n moatje jannever en n sloatje tebak !	
BPE:	n moatje jannever en n slu tebak !	66.0633
Char:	n moatje jannever en n sloatje tebak !	100.

Table 53: Sentences where the BPE encoding outperforms the Character encoding at sentence-level for in-domain GRO→ NL

Form	Example sentences	BLEU score
Ref:	toneelinrichting : een gewone daagse kamer .	
BPE:	toneelinrichting : een gewone daagse kamer	100.
Char:	toneeling : een gewone dagse kamer .	41.1134
Ref:	de dokter pakt naald en draad .	
BPE:	de dokter pakt naald en draad .	100
Char:	de dokter pakt nald en draad .	50
Ref:	sneeuw o wonder nu sneeuw ik onder .	
BPE:	snie o wonder nu sneeuw ik onder .	86.334
Char:	snie o wonder nu snieuw ik onder .	43.167
Ref:	als hij er maar wat aan verdiende !	
BPE:	als hij er maar wat aan verdiende !	100
Char:	snie o wonder nu snieuw ik onder .	59.6949

Table 54: Sentences where the Character encoding outperforms the BPE encoding at sentence-level for in-domain GRO→ NL

Form	Example sentences	BLEU score
Ref:	ze zei geen woord weerom op haar verwijten en hielp vader weer als gewoon .	
BPE:	ze zei geen woord weerom op hun verwijten en hulp vader weer dan gewoon .	48.1098,
Char:	ze zei geen woord weerom op haar verwijten en hulp vader weer als gewoon .	81.5355
Ref:	toen kwam kleinzoon jan naarbinnen , vroeg hoe opoe het nu vond .	
BPE:	toed kwam kwam naar bineen vroeg hoe opnieuw heeft nu vond .	21.9901,
Char:	toen kwam kleenzeun jan naar binnen vroeg hoe opoe het nu vond .	57.5758
Ref:	" laat me ’t schoenmakersvak toch leren , moeder , vader deed dat immers ook ! "	
BPE:	" laat me het schoenmaakersvak .	6.5732
Char:	" laat me het schoenmakersvak toch leren , moeder , vaar deed het ook . "	42.9507
Ref:	twee harten , een koers , ontnuchteren pas later .	
BPE:	twe harten , één koers , ontnuchteren eerst later .	31.7023
Char:	twee harten , een koers , ontnuchteren eerst later .	74.1945
Ref:	hij komt terug , maar hoe hij ook zocht , zijn jas is er niet .	
BPE:	hij komt weerom , maar hoe te zocht , zijn jahze is ter niet .	25.7826
Char:	hij komt weerom , maar hoe hij ook zocht , zijn jas is ter niet .	69.0167

Table 55: Sentences where the BPE encoding outperforms the Character encoding at sentence-level for GRO→ NL Out-domain

Form	**Example sentences**	BlEU-score
Ref:	martha die ik verspeeld had .	
BPE:	martha die ik verspeeld had .	100
CHAR:	martha die k verspeeld had .	48.8923
Ref:	vrijdag , dan is het de negende dag .	
BPE:	vrijdag , dan is het de negende dag .	100
CHAR:	vrijdag , dan is te negende dag .	52.6806
Ref:	en nu gaan jullie allemaal regelrecht naar huis .	
BPE:	en nu gaan jullie allemaal mooi regelrecht naar huis .	70.1688
CHAR:	en nu gaan jullemaal mooi regelrecht naar hoes .	32.4668
Ref:	grote mensen praatten ook altijd over het weer .	
BPE:	groten proten ook altijd over het weer .	64.0675
CHAR:	groten proeten ook altijd over te weer .	31.3241
Ref:	de straat was nu vol leven.	
BPE:	de straat was nu vol geluid.	70.7107
CHAR:	de strade was nu vol geluid.	38.2603

Table 56: Sentences where the Character encoding outperforms the BPE encoding at sentence-level for out-domain GRO→ NL

Form	**Example sentences**	BlEU-score
Ref:	dunne mist kwam over de spoordijk aandrijven .	
BPE:	dunne dook kwam over erspoordiek aandrijven .	27.8900
CHAR :	dunne dook kwam over de spoordijk aandrijven .	75.0624
Ref:	de stijfsel in het kommetje rook naar lauwe melk .	
BPE:	de stijfsel in de kommechie rook naar laten melijk .	28.9978
CHAR:	de stijfsel in te komechie rook naar lauwe melk .	58.7728
Ref:	alweer een handeltje cohen ? ”	
BPE:	weer een handeltje cohn ?	31.6110
CHAR:	weer een handeltje cohen ? ’	61.4788
Ref:	oóó , hij zegt een vies woord . ”	
BPE:	ou , hij zegt een vijs woord .	45.3025
CHAR:	o , hij zegt een vies woord . ’	75.9836
Ref:	en ga jij maar gauw naar huis miert jong , zei hij .	
BPE:	en ga je maar huis naar huis toe m’n jongen , " zei hij vriendelijk .	16.2674
CHAR:	en ga jij maar gauw naar huis toe mijn jongen , ’ vriendelijk .	50.3175
Ref:	vaarwel , graaf zonder hoofd .	
BPE:	verwel , graf zonder hoofd .	43.4721
CHAR:	varwel , graaf zonder hoofd .	80.9107
Ref:	jan knien is er niet meer .	
BPE:	jan knieuw is er niet meer .	70.7107
CHAR:	jan knien is er niet meer .	100