Kevin Winter, MSc MA BSc BSc

Text Style Transfer

Master's Thesis

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Software Engineering and Management

submitted to

Graz University of Technology

Supervisor

Dipl.-Ing. Dr.techn. Roman Kern

Institute for Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Graz, September 2019

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date	Signature

Abstract

Language can be a powerful tool to convey messages, voice opinions, express emotions and with that ultimately also influence people. Writing is just one form of this expression but has become the most important carrier of knowledge since the invention of the printing press. However, there is more to written texts than just its content. Famous authors are not only known by the stories they tell, but rather how they tell it. The way people write, their writing style, is a product of various factors. Ultimately however, it is based on their individual understanding of language and the world. In this thesis we investigate methods that allow for the identification of stylistic patterns in written text and translations between them. Such style transform algorithms could be used to improve readability of complex documents and therefore increase access to otherwise inaccessible knowledge. The methods used are based on stylistic language models and unsupervised word-to-word translations. For evaluation, measures for syntactical and semantic similarity as well as a classifier-based authorship score are used. The results show a significant increase in the authorship score and a significant decrease in the syntactical similarity while maintaining a significantly high semantic similarity for all methods developed, except one.

Contents

List of Figures

List of Figures

List of Figures

List of Tables

List of Tables

List of Acronyms

BLEU Bilingual Evaluation Understudy

CBOW Continuous Bag of Words

CNN Convolutional Neural Network

EOS End of Sentence

GAN Generative Adversarial Network

MUSE Multilingual Unsupervised and Supervised Embeddings

NMT Neural Machine Translation

POS Part-of-Speech

RNN Recurrent Neural Network

XML eXtensible Markup Language

1. Introduction

Language can be a powerful tool to convey messages, voice opinions, express emotions and with that ultimately also influence people. Writing is just one form of this expression, but has become the most important carrier of knowledge since the invention of the printing press. With a global literacy rate of close to 90% (Roser & Ortiz-Ospina, 2019) most people on earth read almost every day. May it be signs to purchase goods, street names to navigate through cities, books of one's favorite author, research papers or just the daily dose of social media. While writing is a mean of transporting information there is more to it than just its content. While street signs leave little room for variation and interpretation, the way a message is written can change how people read it. Famous authors are not only known by the stories they tell, but rather how they tell it. And even research papers, originally designed to purely distribute knowledge, may have different impacts depending on the how they are written.

The way people write, their writing style, is a product of various factors. Ultimately however, it is based on their individual understanding of language and the world. This understanding may be very similar among people that grew up and live in close proximity to each other and may be very different if that is not the case. Dialects are formed, clustering similar uses of language together. Nonetheless, no matter how similar two people may be in regards of these

linguistic features, there will always be slight deviations, giving everyone its own idiolect. McMenamin (2002) defines idiolect to be "the individual's unconscious and unique combination of linguistic knowledge, cognitive associations, and extra-linguistic influences.".

The idiolect of either arbitrary of very specific individuals has been studied by researchers of both linguistics (McMenamin, 2002) and computer science (Juola et al., 2008). Famous examples include the investigation of work of Shakespeare (Hope, 1994), James Joyce's masterpiece Ulysses that arguable was not written by him but rather five other authors (Schoenbaum, 1966) and of course the analysis of the "Unabomber" Ted Kaczynski's manifesto that at last led to his arrest (Crain, 1998). While much research was done in these fields, little is known on how to extract the idiolect of an author and apply its stylistic features to another text.

Such style transform algorithms could be used to improve readability of complex documents and hence also increase access to inaccessible knowledge. The lingo of research papers, medical reports or legal documents could be translated into everyday language. Books could be rewritten in the style of anyone's favorite author, stories could be transformed into songs or songs into poems. In this thesis, methods of modelling and transferring the idiolect of different authors will be investigated.

The main research questions of this thesis hence evolve around the evaluation of an algorithm that adjusts the linguistic style of a sentence while retaining the content. These questions are defined as follows:

R_1 : Is it possible to transform a sentence, such that the style of the sentence approaches the style of a given corpus?

R_2 : Is it possible to retain the semantic of sentences when transforming them?

From these questions we can derive following hypothesis to be tested:

H_1 : The transformed sentences are more similar to the sentences of the target author than the original.

$H_{2.1}$: The transformed sentences are semantically similar to the original.

$H_{2.2}$: The transformed sentences are syntactically different to the original.

2. Background

2.1. Neural Networks

Inspired by the information propagation present in the human nervous system Rosenblatt (1958) invented the first artificial neural network. This so called perceptron perceives input signals and applies a projection to it. Then the signals are accumulated and passed though a threshold function. This mimics the electrical potential that is build up in a neuron until a critical threshold is reached and the neuron fires, forwarding the signal to the neurons connected to its dendrites (Schandry, 2007; Haykin, 1994).

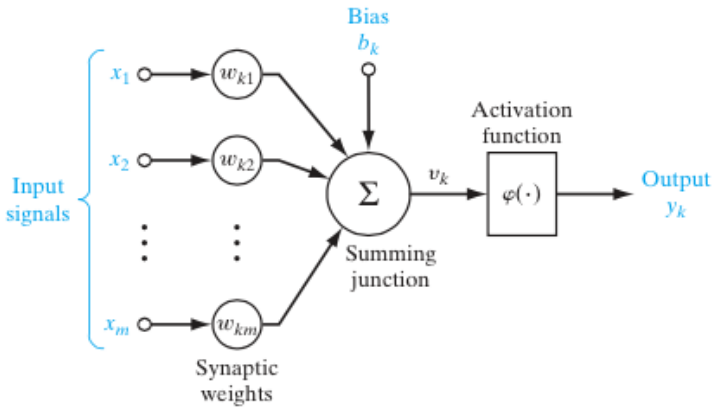

Figure 2.1.: Perceptron Model (Haykin, 2009)

This simple model is, when chosen the right set of projection weights w, capable of any binary classification as long as the two classes are linear separable. The weights can be learned from data using optimization techniques like Gradient Descent, as long as the threshold, also called activation or squashing function, is differentiable. Stacking layers of multiple perceptrons results in a feed forward neural network. In this case, the errors that are observed in the last layer when comparing the output of the network to the desired output have to be propagated back through the network. The back-propagation algorithm does precisely that (Rumelhart, Hinton, & Williams, 1985).

Since the invention of the perceptron many more architectures where defined, following the same principle of combining inputs in a non-linear way and learning the model weights by propagating back the errors. Two fundamental architectural concepts here are Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

2.1.1. Recurrent Neural Networks

In the feed forward neural network inputs are fed to the first layer of the network and all following layers receive only the output of the previous layer as input. Recurrent neural networks contain recurrent connections, connecting the output of the units to the inputs of themselves. The inputs are supplied to the network sequentially, where at each step both the new input and the output of the neuron from the previous step are fed in as new inputs (Elman, 1990). This has the advantages of preserving the order of the sequential features as well as being able to remember information from previous time steps. Additions to this concepts were made by Hochreiter and Schmidhuber (1997) as well as Cho, Van Merriënboer, Bahdanau, and Bengio (2014) by introducing gates that

allow for controlling the information flow inside a neural cell. RNNs are used to model temporal data, which makes them well suited for language modelling. Systems that include RNNs are state-of-the-art in many tasks like neural machine translation (Wu et al., 2016), image-to-text translations (Malinowski, Rohrbach, & Fritz, 2015) and question answering (Kumar et al., 2016).

2.1.2. Convolutional Neural Networks

The basic concept of CNNs is sharing the same weights among neurons. This only makes sense if the inputs connected to these weights may observe the same feature patterns. Then, sets of weights can be interpreted as receptive fields filter maps, that are applied on every part of the input space. LeCun et al. (1989) first used this approach to recognize handwritten digits. Later, more feature maps and layers of convolutions were added to create more complex architectures (LeCun, Bottou, Bengio, Haffner, et al., 1998; He, Zhang, Ren, & Sun, 2015). While CNNs have mainly been used in image processing for many years, they are now used in other fields like time series analysis and natural language processing too. Here, convolutions are usually applied along characters and words, exploiting n-gram based patterns, phrases or word prefixes and suffixes (Bai, Kolter, & Koltun, 2018; W. Yin, Kann, Yu, & Schütze, 2017).

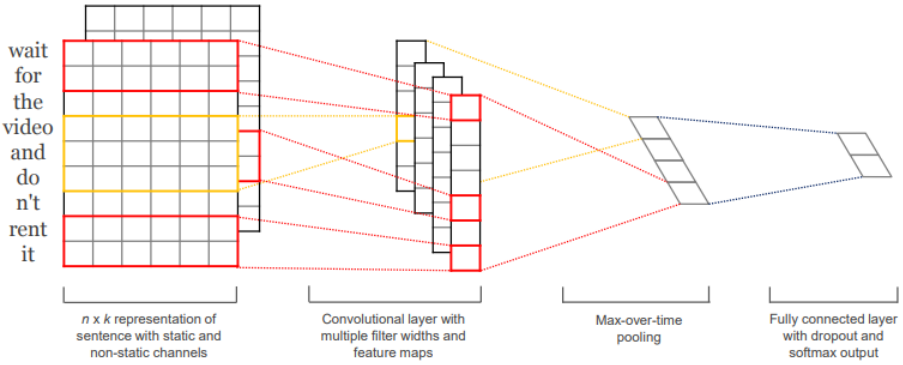

Figure 2.2.: CNN Model for Text Classification (Kim, 2014)

2.2. Language Models

The goal of language models is to learn a representation of certain aspects of language. This can be the joint or conditional probability distribution of words. Within one language, between languages or over multiple languages.

2.2.1. Markov Chains

Andrey Markov, known for his work on stochastic processes, was particularly interested in the independence of future states to previous states of a process (Markov, 1954). Hence he gave the name to the Markov property, which requires a stochastic process to be memoryless. That means that the next state only depends on the present state, and hence the knowledge of the past states has no influence. This is also known as the limited horizon property.

$$P(x_n|x_{n-1}) = P(x_n|x_{n-1}, x_{n-2}, ...x_0) \tag{2.1}$$

Here we use the short notation x_n to denote a random variable $X_n = x_n$. Considering a stochastic process Markovian (the Markov property holds true) has major computation implications since no previous event need to be stored. Therefore Markov processes and Markov chains (Markov processes with district state space) have become popular in many areas of information theory and computer science. In 1948 Claude Shannon used Markov chains to model language as sequence of characters (Shannon, 1948) considering one letter at the time. The same can be applied for sequences of words. In any case, the present state might not be sufficient to model the probabilities of the next state. m-order markov chains reintroduce limited memory of size m.

$$P(x_n|x_{n-1}, ..., x_{x_n-m}) = P(x_n|x_{n-1}, x_{n-2}, ..., x_0) \tag{2.2}$$

2.2.2. Hidden Markov Models

In hidden Markov models, the states themselves can not be observed directly. Instead, other visible variables are observed at each point in time that are emitted by the hidden states. The sequence of observed visible states then gives information about the sequence of hidden states. The parameters of hidden Markov models consist of the start probabilities of each state π, the transition probabilities between states A and the emission probabilities B. These parameters, if not known, can be estimated using maximum likelihood methods like the expectation-maximization algorithm. Then, given a sequence to observed visible states, the most likely hidden state sequence can be decoded using the Viterbi algorithm (Viterbi, 2010). However the time complexity of this algorithm is $O(|S|^2T)$ with $|S|$ being the number and states and T the length of the sequence. For large state spaces, that becomes unfeasible quickly. Alternatively, an approximate result can be found using beam search. This

algorithm uses breadth-first search keeping only β best sequences at each point in time. Hidden Markov models have been used in natural language processing for speech recognition and part-of-speech tagging, with the tags being the hidden states that emit words Jurafsky and Martin, 2009; Manning and Schütze, 1999.

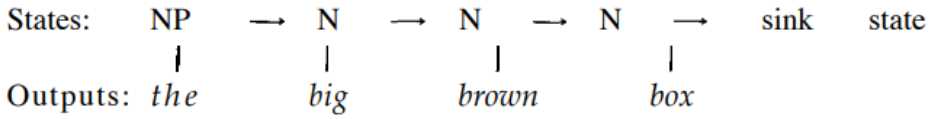

Figure 2.3.: Hidden Markov Model for POS-tagging (Manning & Schütze, 1999)

2.2.3. Neural Language Models

The increasing computational feasibility of neural networks has led to new methods in many fields of computer science, including the language modelling (Bai et al., 2018; W. Yin et al., 2017). In 2003 Bengio et al. already modelled the conditional probabilities of the next word in a sentence given the previous words as a feed forward network with a single hidden layer and a softmax output layer (Bengio, Ducharme, Vincent, & Jauvin, 2003). The input and output words were encoded as one-hot vectors. The hidden layer effectively learned to represent the joint probability of the context words given. While since then, many different architectures have been introduced, the Continuous Bag of Words (CBOW) and the Skip-gram models, both described by Mikolov, Chen, Corrado, and Dean, 2013 have seen received the most attention. In Figure 2.4 the two models are shown.

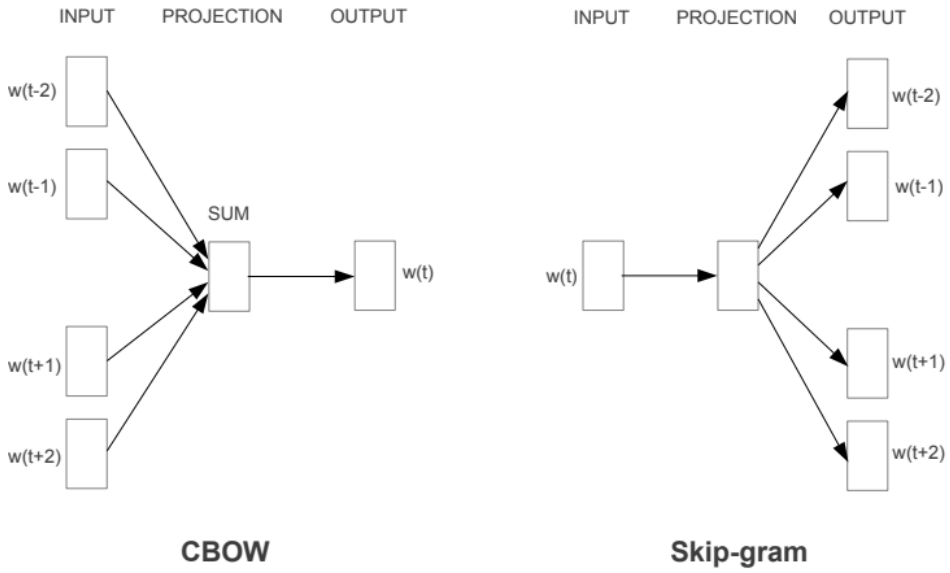

Figure 2.4.: CBOW and Skip-gram Model (Mikolov, Chen, Corrado, & Dean, 2013)

‘ The CBOW model tries to predict a word in a sentence given the words surrounding it. All input words use the same projection and are averaged. Therefore the information of the order of words is lost. The projection uses continuous representations of the context words unlike standard bag-of-words models. In contrast, the Skip-gram model takes one word as input and tries to predict the context words surrounding it. During the evaluation process, the authors noticed that the word representations in the projection layers of the Skip-gram model show certain semantic patterns. Words that appear in a similar context are close to each other. Semantic pairs like countries to cities, countries to currencies, adjectives to adverbs and more share roughly the same vector difference. An example of this is shown in Figure 2.5

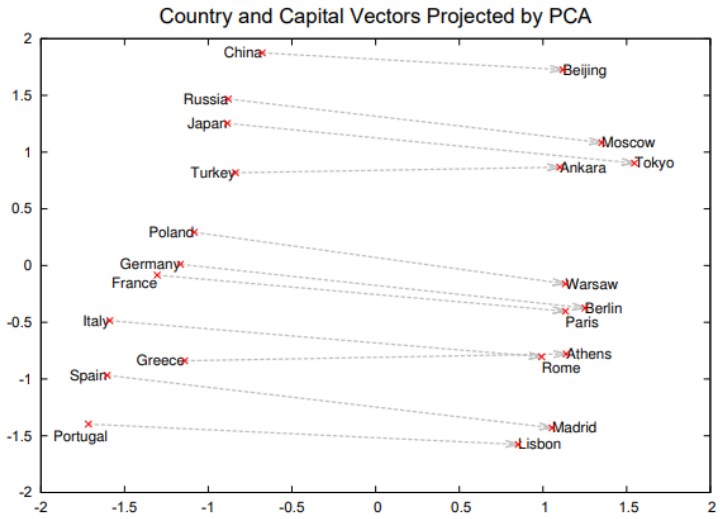

Figure 2.5.: Two-dimensional PCA projection of Skip-Gram vectors (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013)

‘ Additions to this model were introduces to improve the performance, like replacing the computationally expensive full softmax with hierarchical softmax (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). Other models like GloVe (Pennington, Socher, & Manning, 2014) construct a co-occurance matrix of the words in a corpus first and learn a low dimensional representation of that matrix. FastText (Bojanowski, Grave, Joulin, & Mikolov, 2017) learns representations of character n-grams and constructs word representations as a combination of these n-grams. This allows to obtain representations of unknown words. While these approaches learn word representations using the context of these words, in the evaluation phase the context is ignored. Words however may have different meanings depending on the context given. This introduces the problem of polysemy, and even more, homonymity. Arora, Li, Liang, Ma, and Risteski, 2018 showed that the vector representation of words with multiple meanings is located at the superposition of the ideal representations of the individual

meanings of that words. Hence the different meanings are essentially averaged. A more recent approach uses a bidirectional Transformer (Vaswani et al., 2017) architecture to retrieve context aware word representations (Devlin, Chang, Lee, & Toutanova, 2018). Transformer based models mainly use weighted softmax function in order to learn the importance of individual words of the input and output context.

The representations obtained with these methods are used in many applications since they encode more information than other word vectorization methods like bag-of-words or TF-IDF and are continuous. In combination with CNNs and RNNs they achieve state-of-the-art performances in tasks like text classification (Zeng, Liu, Lai, Zhou, Zhao, et al., 2014; Zhang & Luo, 2018) or machine translation (Wu et al., 2016; Vaswani et al., 2017).

2.3. Writing Styles

In computational linguistics, researchers discovered a series of features that are indicative of ones writing style. These features may be grouped in three classes, lexical features, syntactical features and structural features (Zheng, Li, Chen, & Huang, 2006). Lexical features include character-based features like the total number of characters in a text, number of specific types of characters like white-spaces, digits and symbols (e.g. punctuation symbols, smileys) as well as frequencies of letters. Character n-grams are used as well to capture lexical, grammatical and orthographic preferences (Koppel, Schler, & Argamon, 2009; De Vel, Anderson, Corney, & Mohay, 2001). Word-based features are the number of words per sentence or text, word length or vocabulary size. Additional measures exist, that are based on words only occuring once (Hapax legomena) or twice (Hapax dislegomena) or vocabulary richness (e.g. Yule's

K measure, Simpson's D measure) (Tweedie & Baayen, 1998). Word n-grams are used as well. Syntactical feature look at the use of punctuation, frequencies of Part-of-Speech (POS) tags and function words (Zheng et al., 2006; Koppel et al., 2009). POS tags label categories of words with similar grammatical properties. The degree of detail varies depending on what set of POS tags are used. A course set would consist of nouns, verbs, adjectives, adverbs, pronouns, prepositions, conjunctions, interjections and determiners. More detailed versions like the Penn Treebank (Marcus, Santorini, & Marcinkiewicz, 1993) can be seen in figure 2.6.

Table 2:
The Penn Treebank POS tagset

1.	CC	Coordinating conjunction	25.	TO	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subord. conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	`	Left open single quote
22.	RBS	Adverb, superlative	46.	``	Left open double quote
23.	RP	Particle	47.	'	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	''	Right close double quote

Figure 2.6.: Penn Treebank POS Tags (Marcus, Santorini, & Marcinkiewicz, 1993)

Function words are words that express relationships rather than content, like determiner, prepositions and conjunctions (Zheng et al., 2006). Structural features include the length of paragraphs and their separators as well as the use of quotes, indentations and greetings.

These features are commonly used to perform authorship attribution (Juola et al., 2008). So was Argamon, Koppel, Fine, and Shimoni (2003) able to find differences in the writing styles of men and women. The authors stated, that men tended to use more specifiers, describing the content in more detail, adjectives and nouns. Women used more pronouns and shorter words. Since the increased popularity of neural networks CNNs (Zeng et al., 2014; Kim, 2014) and RNNs (Zhang & Luo, 2018) have been used more often for character- or word-based classification.

2.4. Neural Machine Translation

Machine translation is the task of translating text given in one language to another language. Usually this is done by training an algorithm on pairs of sentences in those languages. Therefore, a large parallel corpus is required, consisting of sentences that ideally were translated sentence-wise by professional translators. Sometimes, parallel corpora need to be aligned first, by identifying pairs of sentences. For many language pairs parallel corpora already exist, usually created from translated legal texts, news articles, literature, Wikipedia articles (Schmied, 2019; Tiedemann, 2012; Ziemski, Junczys-Dowmunt, & Pouliquen, 2016).

Translation can be seen as a sequence to sequence task, requiring methods

to generate new sentences from existing ones. Neural machine translation algorithms often model this by creating an encoder-decoder architecture. The encoder translates the source sentence into a latent feature space. The decoder takes latent representations of sentences as input to generate a new sentence in the target language. Both the encoder and the decoder can be modelled using RNNs (Cho, Van Merriënboer, Gulcehre, et al., 2014). Classical RNN can only access information of words before. In order to also get information from future words, bidirectional RNNs are needed, effectively stacking the hidden states of RNNs that go through the sentence in opposite direction (Sundermeyer, Alkhouli, Wuebker, & Ney, 2014). Additionally attention mechanisms can be applied to ease learning translations between languages with different word orders (Bahdanau, Cho, & Bengio, 2014; Luong, Pham, & Manning, 2015; Wu et al., 2016). More recently, transformer networks were introduced, disregarding RNNs all together and using attention mechanisms as main component (Vaswani et al., 2017). What these approaches have in common however, is that they all need a parallel corpus.

Unsupervised methods try to avoid this problem by learning language structures on corpora of the individual languages first and combine them afterwords. Conneau, Lample, Ranzato, Denoyer, and Jégou (2017) train word embeddings on both languages first. The embedding spaces are then aligned by supplying a set of word pairs that are known translations between the two languages. A toy example of this Multilingual Unsupervised and Supervised Embeddings (MUSE) method is shown in Figure 2.7.

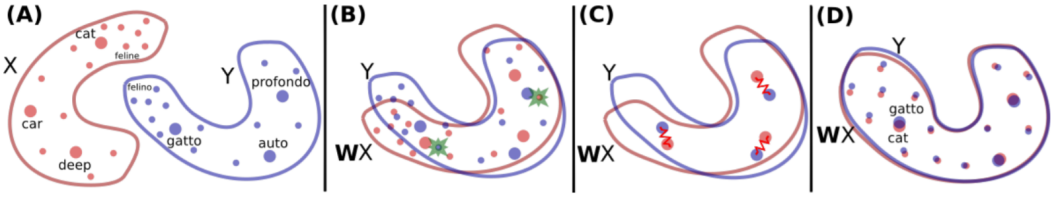

Figure 2.7.: MUSE Method Illustration (Conneau, Lample, Ranzato, Denoyer, & Jégou, 2017)

The resulting vectors form two matrices, that are used as input for an orthogonal Procrustes problem Gower, Dijksterhuis, et al. (2004). The goal of the problem is to find an orthogonal matrix R that maps matrix A to matrix B.

$$R = \arg\min_{\Omega} \|\Omega A - B\|_F \quad \text{where} \quad \Omega^T\Omega = I \tag{2.3}$$

The resulting aligned embeddings can then be used to generate word-by-word translations. A word in the source language is first encoded using the source embeddings. This vector is then transformed into the aligned embedding space using the transition matrix R. Finally, using nearest neighbor search, the closest vector in the target embedding is obtained and the word it corresponds to returned. For measures of distance the cosine similarity is used. This word-to-word translation may at best be a bad approximation of the true translation, depending on the language pair and the sentence structure. In order to obtain a correct the word order, Lample, Conneau, Denoyer, and Ranzato (2017) suggested a denoising auto-encoder. It is trained on sentences that were altered by removing words or changing the order of words. The task during training is to reconstruct the original sentence.

2.5. Neural Style Transfer

The development of style transfer algorithms is not a new idea. While until recently little approaches existed in the field of natural language processing, that cannot be said for image processing.

First approaches used bilinear models to model portraits as a combination of pose and actual facial features (Tenenbaum & Freeman, 2000). As a result, the authors were able to create portraits of people in a new pose. Similarly, the style of a painting can be separated from its subject. This is, because the contours of the subject are usually bigger and hence are composed of lower frequencies than the stylistic patterns used (Hertzmann, Jacobs, Oliver, Curless, & Salesin, 2001; Gatys, Ecker, & Bethge, 2016). Observing the pronounced features of an image at different degrees of detail reveals differnt layers of pattern. Those layers can be produced by simply rescaling the image (Hertzmann et al., 2001) or by exploiting the already hierachical structure of convolutional neural networks (Gatys et al., 2016).

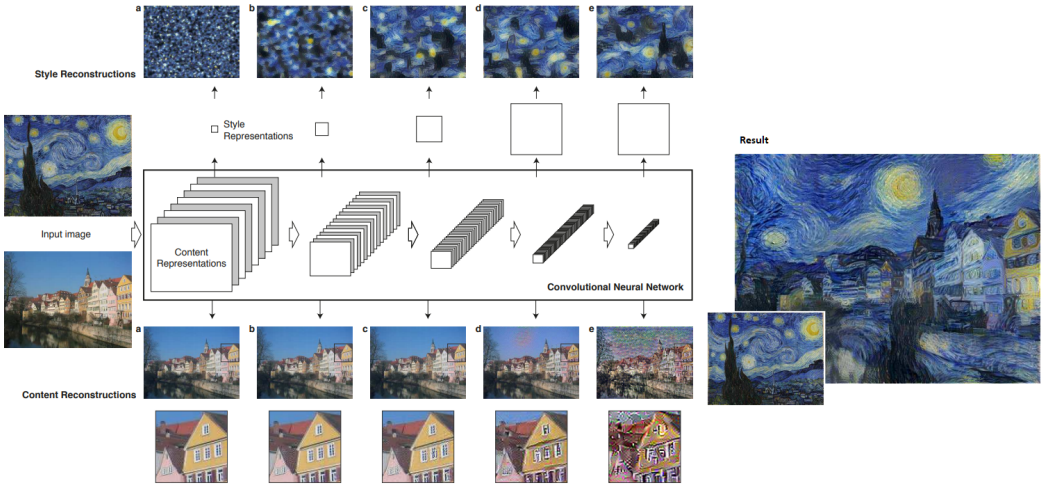

Figure 2.8.: Image Style Transfer using CNNs (Modified from Conneau, Lample, Ranzato, Denoyer, and Jégou, 2017)

By transferring only the more detailed layers to another image sustains the content of an image while changing its style and creating image analogies (Durk P Kingma, Mohamed, Rezende, & Welling, 2014). Most recent approaches use Generative Adversarial Networks (GANs) to perform Image-to-Image translation (Zhu, Park, Isola, & Efros, 2017; Mo, Cho, & Shin, 2018).

3. Related Work

Recently, there has been growing interest in algorithms that are able to generate text. This may be largely due to the raise of neural text generation model and their success in areas like machine translation (Sutskever, Vinyals, & Le, 2014). Since then, sequence-to-sequence models have been creates employing LSTMs (Hochreiter & Schmidhuber, 1997), variational auto-encoders (VAEs) (Diederik P Kingma & Welling, 2013) and more recently generative adversarial networks (GANs) (Goodfellow et al., 2014). What most of those models have in common is the need of a parallel corpus. While there are various such corpora for different languages, this is not the case for different styles. Nonetheless, some exist.

The work of Shakespeare was repeatedly translated into modern English to increase its accessibility. Such translations were aligned and used as training data for style transfer methods (Xu, Ritter, Dolan, Grishman, & Cherry, 2012; Jhamtani, Gangal, Hovy, & Nyberg, 2017). The same was done using different translations of the bible Carlson, Riddell, and Rockmore, 2018 or article titles of different newspapers (Fu, Tan, Peng, Zhao, & Yan, 2018). In these cases, simple statistical language models (Xu et al., 2012) or neural sequence-to-sequence models (Carlson et al., 2018; Jhamtani et al., 2017) were trained in a supervised manner.

Other work on text style transfer without parallel corpora considers various types of styles. Common types are sentiment (Shen, Lei, Barzilay, & Jaakkola, 2017; Hu, Yang, Liang, Salakhutdinov, & Xing, 2017), gender (Prabhumoye, Tsvetkov, Salakhutdinov, & Black, 2018), age (Lample, Subramanian, Smith, Denoyer, Boureau, et al., 2018) and related languages (Yang, Hu, Dyer, Xing, & Berg-Kirkpatrick, 2018). In these cases no real parallel corpora are created. Instead, small domains like movie reviews used to create mono-corpora for each category (e.g. positive and negative, male and female). This data is then used to train auto-encoders with either multiple decoder (Shen et al., 2017), conditioning mechanisms based on disentangled latent representations (Hu et al., 2017; Xi Chen et al., 2016) or constraints (Ficler & Goldberg, 2017; Hu et al., 2018). In addition to the reconstruction error of the auto-encoder, adversarial methods are used in combination with classifiers to enforce the stylistic similarity (Hu et al., 2017; Shen et al., 2017; Fu et al., 2018). (Hu et al., 2018) trained a sequence-to-sequence model with reinforcement learning to fill in blanks in sentence templates. Some also use back-translation to obtain a latent representation of sentences using the encoder of pre-trained neural translation models (Prabhumoye et al., 2018; Lample et al., 2018).

The evaluation measures used are also not consistent. Where parallel corpora exist, the target sentences are known. Therefore BLEU score is used to measure the difference to the ground truth (Xu et al., 2012). Interestingly, other authors measured the BLEU score between source and generated sentence, assuming a high value would indicate good quality and fluency (Yang et al., 2018; Shen et al., 2017). These authors also measured the perplexity of the target sentence as measure of quality. The systems employing a classifier used its accuracy as measure for stylistic similarity (Shen et al., 2017; Hu et al., 2017; Fu et al., 2018; Yang et al., 2018). In order to evaluate how well the

original content was retained, (Fu et al., 2018) computed the cosine similarity between source and generated sentence.

4. Methods

4.1. System Description

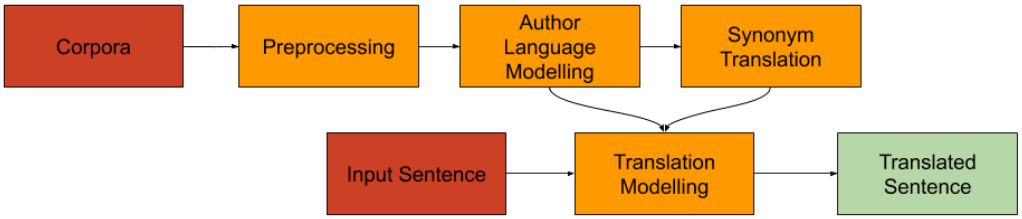

Figure 4.1.: System Overview

4.1.1. Preprocessing

The raw text of the authors we want to model may contain noise, meaning information that is not only irrelevant to our purpose but may also decrease the performance of our models. Therefore the text is first cleaned by removing special characters, lines breaks and numbers. The only exception to this are punctuation marks, `.`, `,`, `!` and `?`, since the use of them may be unique to each author. Additionally, all upper case characters are converted to their

lower case equivalent in order not to treat words at the beginning of sentences differently.

4.1.2. Author Language Modelling

The preprocessed sentences are then used to create language models for the individual authors. The model used developed in this thesis contains three submodels. First, the transition probabilities between words are modelled. For this a third order Markov chain is used in order to capture contextual dependencies. This means, that given three words the model outputs a probability for each word being next. Second, a set of stylistic features is extracted from each word in the corpus. Pairs of words and their corresponding feature vectors are then used to create a probabilistic mapping from feature vector to words that emitted this vector. Third, after translating the corpus to sequences of feature vectors, we compute the transition probabilities between them. For that, a third order Markov chain is used again. This is supposed to capture the stylistic patterns of authors independently of the actual words and hence the content of the sentences. The combination of these language models can be seen as a hidden Markov model with the hidden states being words emitting style feature vectors or the other way around. Since we have transition probabilities between both the words and the feature vectors it is actually more general Bayesian network.

Stylistic Feature Extraction

Following the literature, we saw that there are various features that can be observed when comparing the linguistic stylistics between authors. Some operate

on a character level, mainly in the form of n-grams at different positions in words, like prefixes or suffixes. Others look at individual words, which as a whole represents the authors vocabulary. The length of words, their lexical frequency, the frequencies of certain classes of words, like POS tags or stop words may give some indications. Further, the structure of sentences with regards to their length, complexity and word dependencies are considered.

The models described above require features to be extracted from individual words. This partially excludes the explicit use of sentence structures. Also, the number of features can not be too extensive. This is, because the corpora of individual authors are usually rather small, leading to little statistical evidence for individual combinations of feature vectors if these vectors have a high cardinality. Considering this, the following features are chosen: The POS tag, length and lexical frequency of word as well as whether it is considered a stop word.

Word	POS	Stop Word	Length-Group	Frequency-Group
It	PRP	1	0	2
is	VBZ	1	0	2
undoubtedly	RB	0	2	1
a	DT	1	0	2
variation	NN	0	1	1

Table 4.1.: Spatial Features of an Example Sentence

In Table 4.1 the extracted features of a sample sentence are shown. Both the lengths and the lexical frequencies are binned into three bins representing low, medium and high, again to reduce the cardinality. With the preprocessing step in mind, there can still be around 40 different POS tags of the Penn Treebank

and up to 720 different combinations. In practice however this is not the case, since certain combinations never occur. Nouns are usually not stop words and most stop words have a high frequency since this is why they are considered stop words in the first place. The features are than concatenated into one feature vector, encoded as so called style tokens.

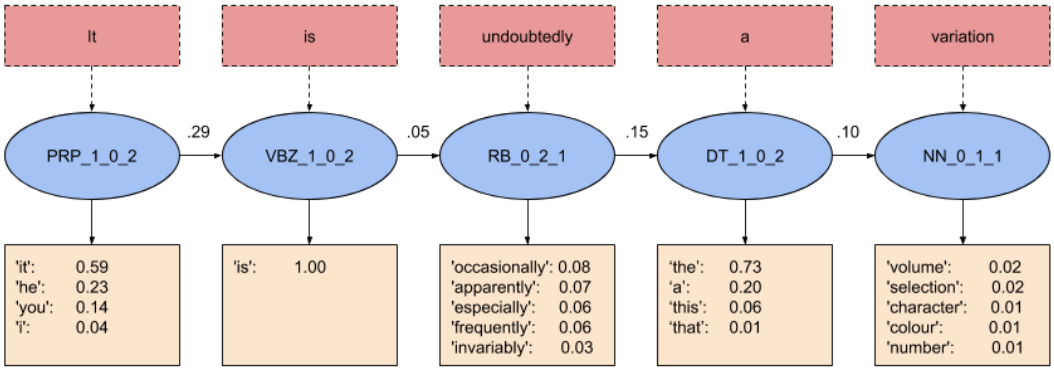

Figure 4.2.: Style Token Emission Probabilities

In Figure 4.2 the extraction of the feature vectors with the resulting probabilities are shown. It becomes apparent that for certain combinations of features the emission space is quite limited.

4.1.3. Synonym Translation

One major aspect that may distinguish one's favourite author from authors is the their vocabulary. The use different words and phrases can change the reading experience even if the core message of a text stays the same. On a word by word basis this essentially describes synonyms. Words with similar meanings given a certain context. This part of our system can find synonyms

that a given author would most likely use. To achieve that, an approach from Neural Machine Translation (NMT) is utilized. NMT The texts of different authors are considered to be different languages and the search for synonyms is framed as a word-by-word translation from one language to another. Most NMT methods are supervised and require parallel corpora in order to learn the relationship between languages. In our case this is not an option. However there are also a few unsupervised methods as mentioned before. The one used here is the MUSE method by Conneau et al., 2017. The method requires word embedding spaces for each language. These are trained using the Word2Vec CBOW (Mikolov, Sutskever, et al., 2013) algorithm with a window size of five words. A vector size of 100 is used since it is big enough to capture the relevant information (Z. Yin & Shen, 2018) and small enough to obtain an expressive mapping between the embeddings. This is because MUSE requires a set of known mappings from one language to the other. These pairs are the anchors for the Procrustes analysis, which is performed to find transformation matrices that align the two different embedding spaces. Before outlining the choice of anchor words, an overview of the synonym translation system is shown in Figure 4.3.

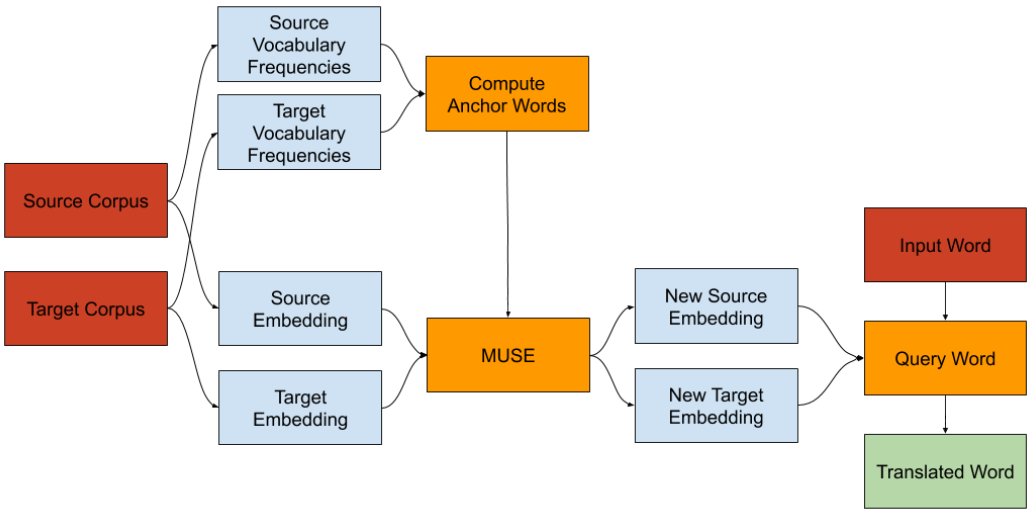

Figure 4.3.: Synonym Translation System

Anchor Words

In the case of two different languages anchor word pairs can be obtained easily by looking up translations in dictionaries. If both the source and target language is English this is not the case anymore. Hence, a different approach is needed for choosing these pairs. Since it is the same language, a simple assumption would be that the words in both are the same and hence any set of words could be used. However, different authors use different vocabularies so that not every word may be present in both languages. Additionally, just because authors may use the same word does not mean that these words are equal in both their languages. One might use it more frequently than the other. Ideally, we want so select words that both authors use similarly frequent and still are common words of the English language itself.

$$W_A = \{w \in V_S \cap V_T | min(P_S(w), P_T(w)) > e \wedge |P_S(w) - P_T(w)| < d\} \quad (4.1)$$

The anchor words W_A are chosen from the common set of words in the vocabularies of the source author V_S and the target author V_T such that the difference in there relative lexical frequencies $P_S(W)$ and $P_T(W)$ is smaller than some difference threshold d and the words are more common in the authors' corresponding vocabularies than e.

Word Queries

The resulting anchor words are used to obtain the new aligned embeddings. In order to find synonyms for a word this word is encoded as a vector using the source embeddings. Since the embeddings spaces are aligned, the vector would point to the same word in the target embeddings space, if the languages would be identical. Due to their differences however, this is not always the case. Performing a nearest neighbor search given the vector one can select one of the closest words found as synonym. To introduce randomness to this process, noise is added before performing the search. This allows for variations in the output sentence, if multiple words are close to the transformed vector in the target embedding space. Figure 4.4 illustrates this method.

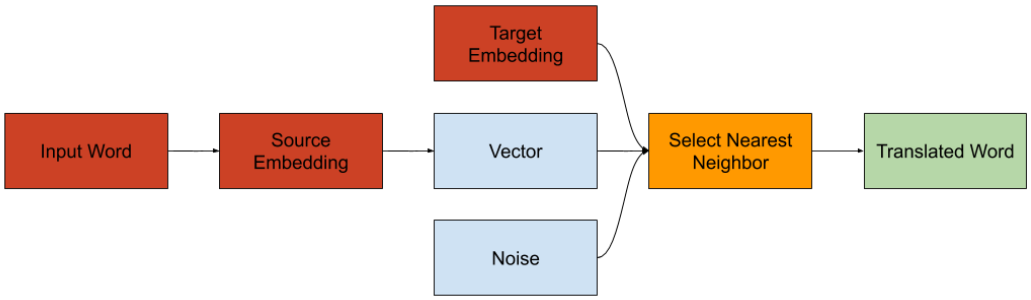

Figure 4.4.: Word Query System

Given this synonym translation it is already possible to obtain word-by-word translations between authors. Varying the amount of noise added, the new sentence will be more or less similar to the original sentence. The meaning of the sentence should also stay intact since words that are similar in the embedding spaces should be similar in meaning too. However, the abstracting capabilities are very limited since the structure of the sentence stays the same. This is, why we introduce a generative approach that merely utilizes this word-by-word translation as one step.

4.1.4. Sentence Search

Now that the main components of our system were defined, they can be connected to create said generative model. The idea is to use the different language models to guide and normalize the generation of a new sentence. This can be seen as decoding problem, common in NMT tasks and usually solved with beam search. The obvious advantage of beam search over other methods being that only the top β most probable candidates are kept track of at every point in time. In this case the points in time are the sequential words in the

generated sentence. In this thesis we investigate two different strategies for decoding, calling them beam search and slot-filling beam search.

Beam Search

When performing beam search we need to compute a score for each candidate and combine these score for each temporary sequence. In our case, the score is the probability of candidate word given the n words before. In Equations 4.2 to 4.4 we see, that this probability consists of a weighted sum of three components, the context $P(w_t|W_c)$, the word language model $P_{LM}(w_t|w_{t-1:t-n})$ and the style token language model $P_{SLM}(w_t|w_{t-1:t-n})$. As input, the method uses these models, the weighting terms and the context words. The context defines the words that may be part of the sentence. We insert the word-by-word synonym translation as context.

$$P(w_t|w_{t-1:1}, \theta) = \lambda_c P(w_t|W_c) + \lambda_{LM} P_{LM}(w_t|w_{t-1:t-n}) + \lambda_{SLM} P_{SLM}(w_t|w_{t-1:t-n}) \tag{4.2}$$

with

$$\theta = \{w_{t-1:1}, W_c, \lambda_c, \lambda_{LM}, \lambda_{SLM}\} \tag{4.3}$$

and

$$\lambda_c + \lambda_{LM} + \lambda_{SLM} = 1 \tag{4.4}$$

For the word language model, the output probability for each word is just the conditional probability of the word in the Markov chain model given the previous n words or $t-1$ words for the first words where $t <= n$. At each point in time the style tokens of the previous words are extracted as mentioned before. For this sequence to tokens the probabilities of the next

token $P_{SLM}(token_t|token_{t-1:t-n})$ are evaluated using the style language model. Given the resulting tokens, the word emission probabilities are weighted and summed as formalized in Equation 4.5.

$$P_{SLM}(w_t|w_{t-1:t-n}) = P_{SLM}(w_t|token_t)P_{SLM}(token_t|token_{t-1:t-n}) \tag{4.5}$$

The context words are added as a third layer, essentially boosting the words of the input sentence. However not all words in a sentence are equally important to be kept. While nouns and especially proper nouns should still be the same in the resulting sentence, the same cannot be said for stop words. This is why we introduce a simple attention mechanism that weights context words in accordance to their POS tags.

$$P(w_t|W_c) = \frac{attention(pos_t)}{\sum_{j=1}^{|W_c|} attention(pos_j)} \tag{4.6}$$

For this the Universal POS tags were used, because of the lower granularity. The weights used can be taken from Table 4.2. These weights follow the heuristic, that the main content of a sentence is captured by the relationship between nouns and verbs. Proper nouns like names should not change at all, while adjectives may be omitted or replaced.

POS tag	weight
PROPN	5
NOUN	4
VERB	3
ADJ	2

Table 4.2.: Context Wort Attention Weights

The probability of a sequence is then just the weighted product of the probabilities of each individual word in that sequence. To avoid the problem of numerical underflow, the implementation uses the sum of log probabilities instead. While this already describes the main algorithm, two additions are added in order to improve the stability of the model. First, because we do not specify a fixed sentence length, sentences of different length may be compared to each other when choosing the best ones. Since all probabilities are less than one, longer sequences will have smaller probabilities than shorter ones. Hence, length normalization is required. While various approaches exist already, the best results were achieved with this novel approach, defined in Equation 4.7.

$$\lambda_{length} = \frac{\beta - 1}{N^2} t^2 + \frac{2(1-\beta)}{N} t + \beta \tag{4.7}$$

We define a concave function that has its maximum of 1.0 at a target length N. From there it decreases symmetrically in both directions, reaching the value of β at length 0 and $2t$. This way, it is easy to adjust the preferred mean output length as well as the deviations from it.

Second, the language models also incorporates an End of Sentence (EOS)

token. To be able to control the length of a sentence, it is necessary to adjust the probability of that token. Therefore, a damping term is introduced, reducing the probability of the EOS token quadratically around the target length N.

$$\lambda_{EOS} = \frac{t}{N^2} \tag{4.8}$$

Slot-Filling Beam Search

The second approach used is roughly equivalent to the first one. However, in this case the style language model is first used to create a sequence of style tokens. For this, we sample from the transition probabilities to find a likely sequence until we encounter an EOS token. Intuitively this sequence should represent typical stylistic patterns an author uses. The idea now is, to use this encoded sentence as a template for the real sentence. This can be seen as a hidden Markov model with the style tokens being the emitted visible states and the actual word of the target sentence being the hidden states. Since we have both the transition probabilities and the emission probabilities, we could find the hidden states using the Viterbi algorithm. However, the state space, which in our case is the author's vocabulary, is far to big to keep track of all possible sequences. Therefore beam search is used again to only keep the most promising candidates. Like in the algorithm described above, we join the emission probabilities with word transition and context probabilities with the same weighted sum.

4.2. Evaluation Methods

The evaluation of the algorithms described in this thesis requires both a special test data as well as measures that can capture syntactical, semantic and stylistic similarities. For this purpose we will utilize the preexisting work of well known authors and use both established and custom made measures.

4.2.1. Sample

In this thesis we focus on the stylistic translation of English prose. Hence we have to exclude famous pieces of poetry and drama. Also, many pieces of literature are still protected by copyright. Luckily, Project Gutenberg ("Project Gutenberg," 2019) collected over 59.000 books so far, offering them without restrictions. Typically these are books whose copyrights expired at some point. With access to this corpus, we want to select authors that created as much text as possible in order to be able to learn adequate models of their writing styles. While the website of Project Gutenberg offers some browsing and filtering functionalities, it does not cover our requirement. Therefore we used a tool called GutenTag (Brooke, Hammond, & Hirst, 2015). This tool allows to filter by genre, language, publication date and country or author attributes like name, nationality, dates of birth and death and gender. Our dataset consists of English prose text published after 1800. This way we want to exclude old English words and phrases in order to be able to focus on stylistic differences given roughly the same vocabulary. The resulting documents were downloaded in eXtensible Markup Language (XML) format. An analysis of the author with the most documents written lead to the selection of the authors shown in the following Table 4.2.1.

	$\|D\|$	$\|V\|$	$\|V_{lemma}\|$
Chesterton	34	36362	23362
Churchill	68	41129	26876
Darwin	28	51073	32255
Dickens	61	57299	32335
Twain	141	54688	33929
Verne	35	48837	32598
Wells	43	53757	32793

Table 4.3.: Descriptive Statistics of Author Corpora

In the columns, we see the number of documents $|D|$ as well as the size of the vocabulary $|V|$ the individual authors used. Defined by $|V_{lemma}|$ is the size of the lemmatized vocabulary. For this the lemmatizer provided in the Python library SpaCy (Honnibal & Montani, 2017) was used. For each of these authors corresponding word, style and emission language models are created. Additionally, all documents found using GutenTag are used to create author independent word embeddings. The style transformation will then be evaluated on a sample of 1000 sentences from this general corpus.

4.2.2. Variables

In the first chapter, the core research questions were defined. The hypotheses derived from them require quantifiable measures to allow for the testing of significance.

Syntactic Similarity

The syntactically similarity as used in hypothesis $H_{2.2}$ is operationalized by the Bilingual Evaluation Understudy (BLEU) Score (Papineni, Roukos, Ward, & Zhu, 2002). It was introduced to evaluate the quality of translations, given a correct translation of the original sentence. Usually this correct translation was performed by linguists or professional translators. The BLEU score measures the mean percentage of matching n-grams between the correct translation and a candidate sentence. The length of the n-grams is increased up to a set maximum N, usually 4. The BLEU score is than computed as the geometric mean of the match scores for each n-gram length.

$$BLEU_N = \left(\prod_{i=1}^{N} \frac{|NG_i(correct) \cap NG_i(candidate)|}{|NG_i(correct)|} \right)^{1/N} \tag{4.9}$$

In Equation 4.9 the formula of the basic BLEU score without normalization is shown. Here, NG_i returns the i-grams of a sentence up to length N. This version however as various downsides as described by B. Chen and Cherry (2014). For example, the score of a candidate is 0 if one of the i-gram scores is 0. This is especially problematic for higher N and shorter sentences. Therefore we use the from the authors introduced smoothing technique number 7, which handles the just mentioned 0 elements and applies smoothing along n-grams adjacent in the sentence.

In our evaluation we want to see low BLEU scores, meaning less exact matches of n-grams while retaining high semantic similarity scores, such that the meaning of the sentence can be considered close to identical.

Semantic Similarity

The semantic similarity as used in hypothesis $H_{2.1}$ is operationalized as follows. We assume that meaning of a sentence can be approximated by the sum of the meanings of the words it contains. Following this approach, some loss is inevitable due to ignoring the word order and sentence length. However, finding the meaning of phrases and sentences as a whole is not trivial. In contrast, we already have a model that models the meaning of words. The word embeddings trained for the translation system map words to vectors in a continuous space. We encode the meaning of a sentence as the average word vectors computed by applying the embedding model E to each of the n words in the sentence x.

$$v(x) = \frac{1}{n} \sum_{i=1}^{n} E(w_i) \tag{4.10}$$

The vectorized sentences can now be compared using a suitable similarity measure. For this, correlation coefficients, the inverse of distance or divergence measures like the euclidean distance could be used. Especially for the comparison of vectors however, the cosine similarity has nice properties. It maps the similarity to range of -1 to 1 with -1 representing vectors that point exactly in the opposite direction, 0 meaning they are orthogonal to each other and 1 representing vectors of same direction. Computationally this measure is also interesting, since it is just the dot product of the two normalized vectors.

$$cos(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||} \tag{4.11}$$

The semantic similarity is hence defined as shown in Equation 4.12, with T being the style transformation algorithm returning the new sentence.

$$s_{sem} = cos(v(x), v(T(x))) \tag{4.12}$$

Authorship Score

The main aspect of this thesis is the stylistic individuality of different authors. Hypothesis H_1 requires a measure of stylistic similarity to a given author. To accomplish that we create models, that can distinguish sentences of our known authors from sentences of a general corpus. This binary classification task is modelled using the CNN architecture used by Zhang and Luo (2018). Convolutional classifiers like these are commonly used in text classification and hence also authorship attribution. In Figure 4.5 the schematics of this architecture is shown.

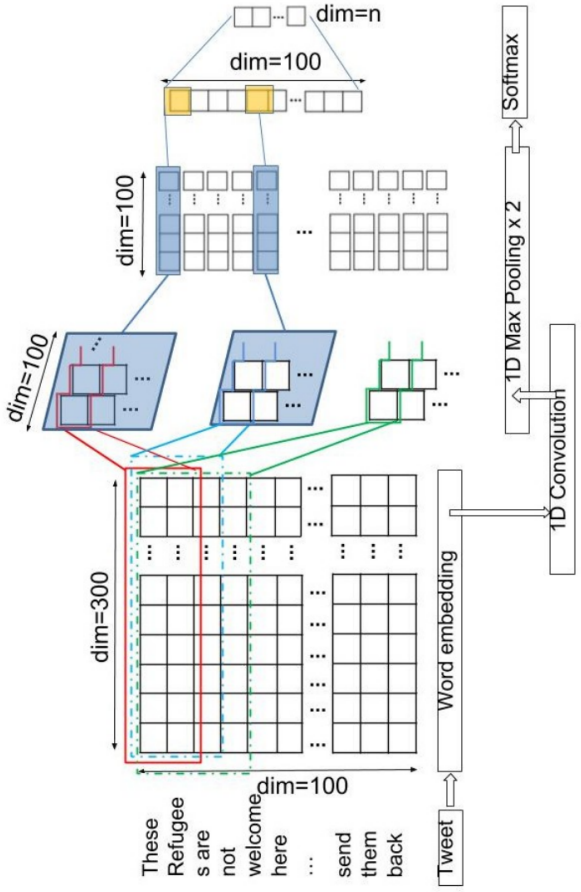

Figure 4.5.: CNN-based Text Classification Model (Zhang & Luo, 2018)

The main idea here is that use of convolutions of different sizes along adjacent words in a sentence. Window sizes of to two to five words can be seen as analysis of 2-grams to 5-grams respectively. The words are first mapped to vectors using word embeddings. In the second direction the filters spread over all dimensions of the word vectors. The resulting filters are than flattened using max-pooling. The last layer is fully connected following a sigmoid function to output values between 0 and 1. These values can be interpreted of probabilities that a given

sentence is written by an author or as a similarity of that sentence to sentences of that author. Therefore, we expect that this values generated by our classifier C will be higher after our transformation T than before.

$$C(T(x)) > C(x) \tag{4.13}$$

4.2.3. Analysis Plan

For each of the seven authors the same 1000 test sentences, randomly sampled from the general corpus of English prose literature are transformed using the following six variations of the style transform methods described in this thesis:

Translate (noise = .2) Synonym Translation only with noise added to the query vector before performing the nearest neighbor look-up. The noise is normally distributed with a mean of 0 and a standard deviation of .2.

Translate (noise = .3) Same, but with noise normally distributed with a mean of 0 and a standard deviation of .3.

Translate (noise = .4) Same, but with noise normally distributed with a mean of 0 and a standard deviation of .4.

Translate (noise = .5) Same, but with noise normally distributed with a mean of 0 and a standard deviation of .5.

Beam-Search Beam Search as described in Chapter 4.1.4 with the output of `Translate (noise = .3)` as context input.

Slot-Filling Beam-Search Slot-Filling Beam Search as described in Chapter 4.1.4 with the output of `Translate (noise = .3)` as context input.

For hypothesis H_1 we perform a paired two-sample t-test because the authorship score is measured before and after the application of the transformation method

and the two measures can be linked to the same sentence ($H_0 : C(T(x)) \leq C(x)$). Hypotheses $H_{2.1}$ and $H_{2.2}$ are tested using a one-sample t-test since we only measure the corresponding variables once. The semantic similarity is tested against a threshold of .9 ($H_0 : s_{sem} < .9$). The syntactical similarity difference is tested against the threshold .5 ($H_0 : s_{syn} > .5$). For all tests p-value is chosen as inference criterion with an α-error limit of .1% ($p < .005$).

The beam search methods may fail to produce a valid sentence because of malformed conditional probability distributions. The resulting incomplete, short sentences are filtered out before the analysis is performed. The true sample size may therefore differ for each method.

In addition to the analyses performed across all authors, the same will be performed on each individual author in an exploratory fashion. The results including the p-values will be reported in the appendix. Note however, that due to the additional testing leads to the problem of α-error accumulation, which is why no statements of significance can be made for these results.

5. Results

The results of this thesis are presented in three parts. In the first part, various examples of sentences transformed using the methods introduced before are presented. Then, exploratory analyses are performed, evaluating the extracted anchor words and the distribution of the evaluation measures. In the last part of this chapter, the results of the confirmatory analysis are presented.

In table 5.1 transformed sentences are shown next to the original. Methods are indicated as Translate (T), Beam-Search (BS) and Slot-Filling Beam-Search (SFBS).

Method	Original:	Translated
T	Her appearance was enough to send a friend into **ecstasies**, or drive an enemy to **despair**.	Her appearance was enough to send a friend into **dreams**, or drive an enemy to **anguish**.
T	How this is done depends upon the **talent** and **cultivation** of the family.	How this is done depends upon the **capacity** and **production** of the family.
T	I did it in fun, said **Jonas**, beginning to see that he had need to be **prudent**.	I did it in fun, said **Kurt**, beginning to see that he had need to be **reasonable**.
SFBS	Did say was that each one of us was to bring fifteen pennies.	Farthings was saying one did bring us that thirty each to of.
BS	As an admirer of newton he endeavoured to teach the ladies to discuss the theory of light.	Teach her the purely arbitrary language in common use to hold wine, and hung out in the light of ladies discuss theory endeavoured snawley as an his little a expression.
SFBS	I do not know about good or evil.	Know do good unselfish not around I or man to.

Table 5.1.: Example result sentences by method

5.1. Exploratory Analysis

5.1.1. Anchor Words

One central element of the methods described in this thesis is the custom application of the MUSE algorithm. This step required the selection of anchor words, linking the embedding spaces of source and target author. This selection was done considering the relative frequencies of word in the vocabularies of both authors. Two parameters were introduced: The minimum relative frequency e and the maximum relative difference d. In the following figures the impact of varying these parameters is shown. As source corpus the general corpus discussed before was used. The higher e is chosen, the more words could be obtained as anchor words. However, this increases the risk of the model not being able to generalize anymore, matching every word with itself. In Figure 5.1 we see that over 1500 words have a relative frequency higher than $1e-6$ in both vocabularies.

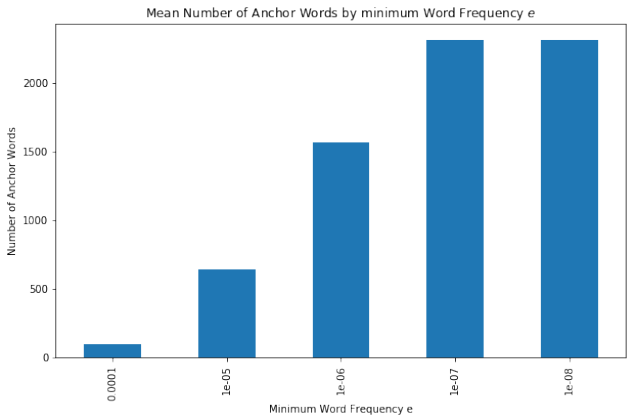

Figure 5.1.: Mean Number of Anchor Words by minimum Word Frequency e

Good anchor words should be equally frequently used by both authors. Therefore, the tolerance of difference between authors should be as small as possible. However, changes in relative frequencies may deviate easily, considering different vocabulary sizes and individual differences due to small corpus sizes. In Figure 5.2 we see the impact of increasing the tolerance up to 0.3.

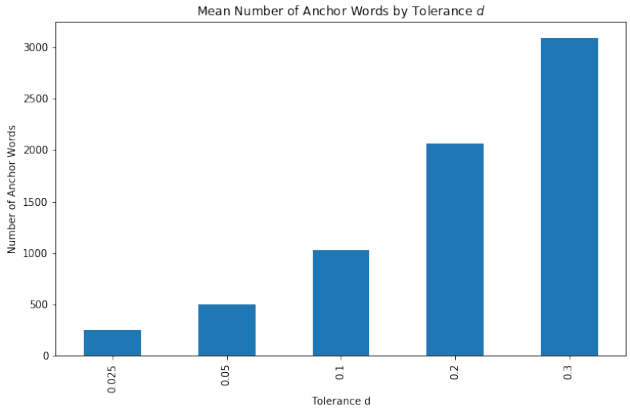

Figure 5.2.: Mean Number of Anchor Words by Tolerance d

Based on these results, a minimum relative frequency of $e = 5e - 10$ and a tolerance of $d = 0.1$ was used. Interestingly, that led to very different numbers of anchors words for each author. A comparison is shown in Figure 5.3

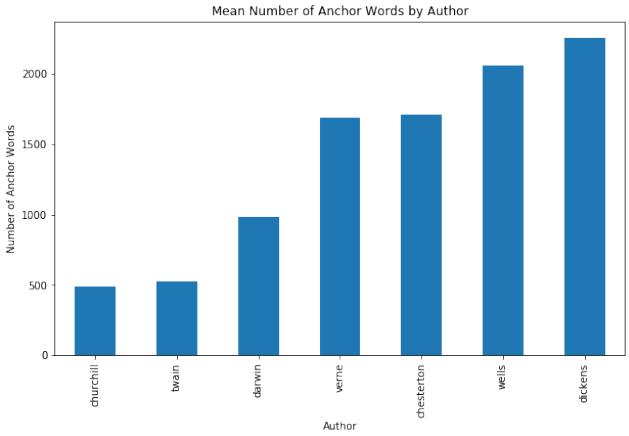

Figure 5.3.: Mean Number of Anchor Words by Author

Given these parameters, Charles Dickens has more than four times as many words in common with an average prose text than Winston Churchill. So either Dickens has a very big vocabulary or a very generic use of words.

5.1.2. Variable Distribution

In this section a detailed comparison between the methods used is presented. The three evaluation measures shown are the authorship score, semantic similarity and syntactical similarly. In Figure 5.4 the change in authorship score before and after applying the transformation method can be observed.

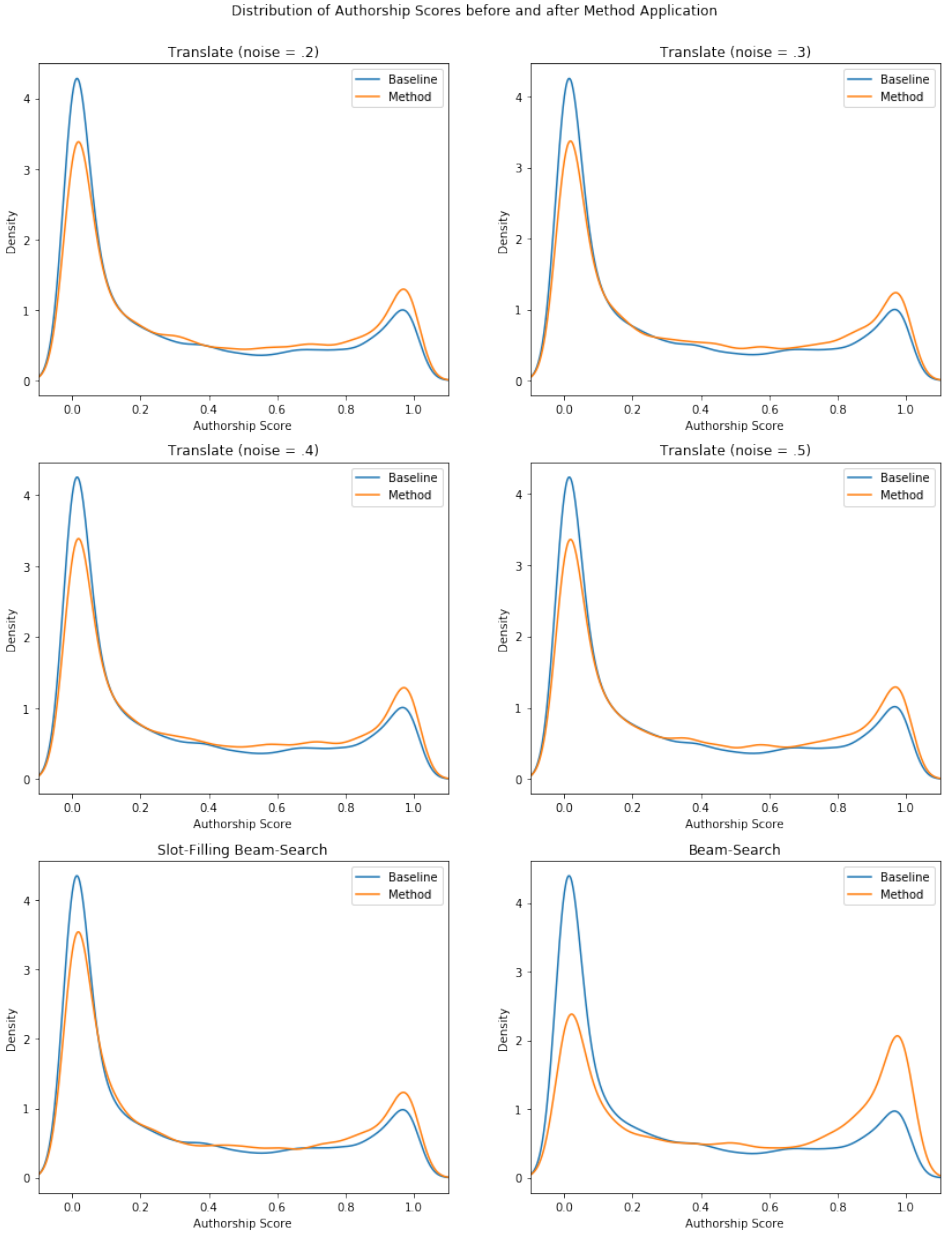

Figure 5.4.: Distribution of Authorship Scores before and after Method Application

The upper four figures present the results of using the MUSE based approach only. Varying the amount of noise introduced has little impact on the amount of change. As one would expect, the most change can be observed with sentences that had a low authorship score originally. The same is true for the search based methods. Only the regular beam search shows noticeably more change than the other methods. In Figure 5.5 the differences per method are visualized explicitly.

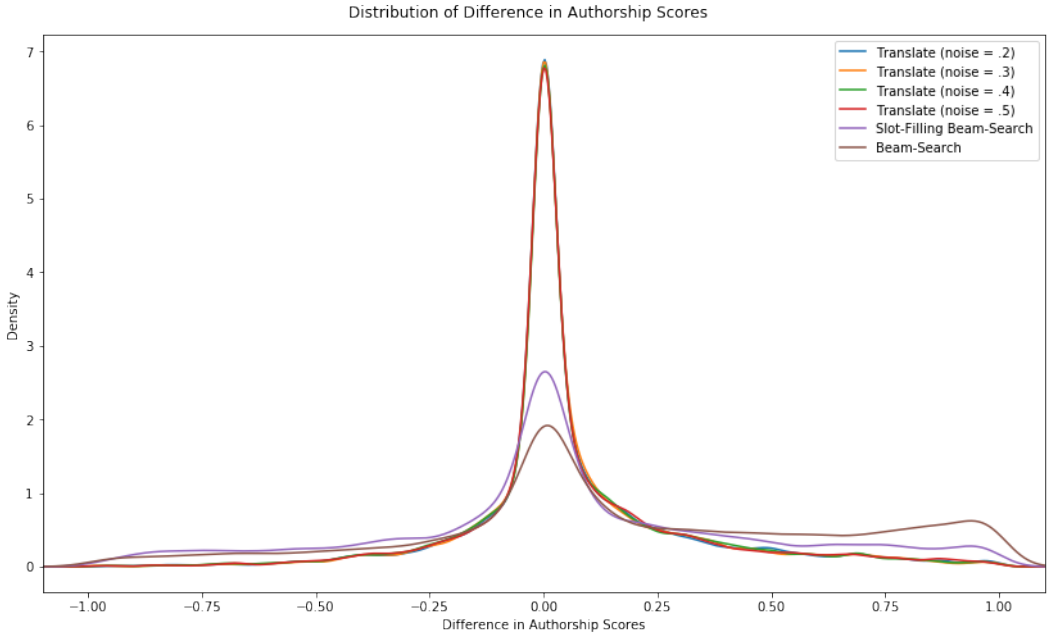

Figure 5.5.: Distribution of Difference in Authorship Scores

The mode of the distributions seems close to zero. However, especially the search based methods show a heavy right tail. In accordance with the small changes in authorship score, semantic similarity is close to 1 for the translate methods, as presented in Figure 5.6. The slot-filling beam-search shows similar

results, potentially retaining larger amounts of the meaning of the sentence. Regular beam-search deviates largely, raising the question if the transformed sentences can still be considered to have the same content.

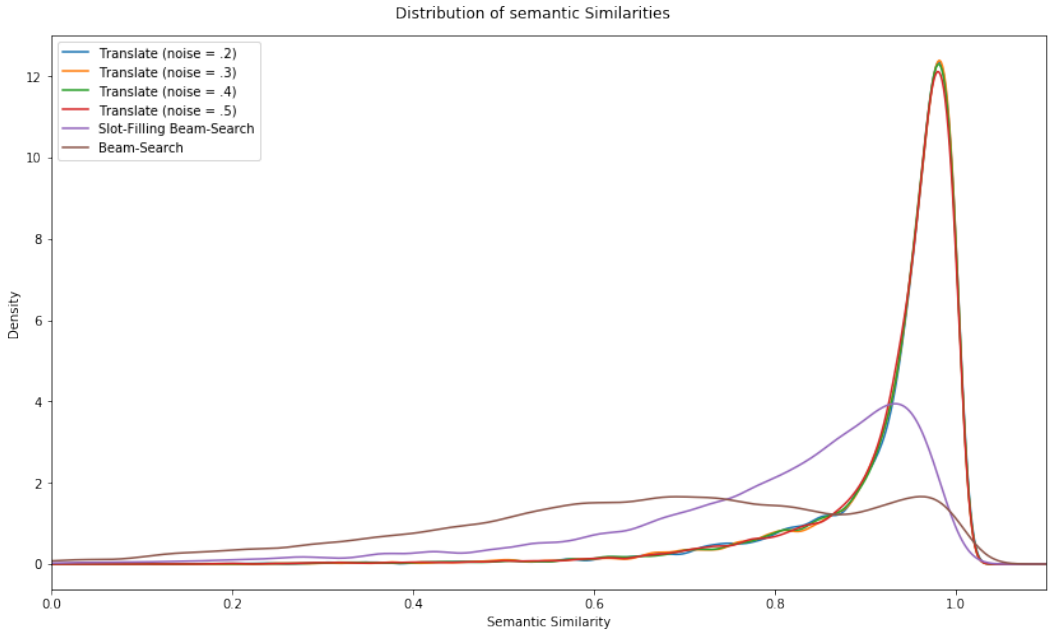

Figure 5.6.: Distribution of semantic Similarities

How much of the original sentence is matched exactly by the new sentence is measures using BLEU scores. Considering the authorship and semantic similarity scores, we might expect the syntactical similarity of sentences transformed using the translate methods to be close to 1 too. This is not the case though. In Figure 5.7 we can see that most sentences translated differ greatly from the original. The translation based approaches barely differ. This could be explained by the assumption, that the random noise added mainly changes words that have no definitive mapping in the target embedding space. Hence, a

change in the noise level up to a certain level still only changes the same words, which results in the same BLEU score, even though the sentences differ.

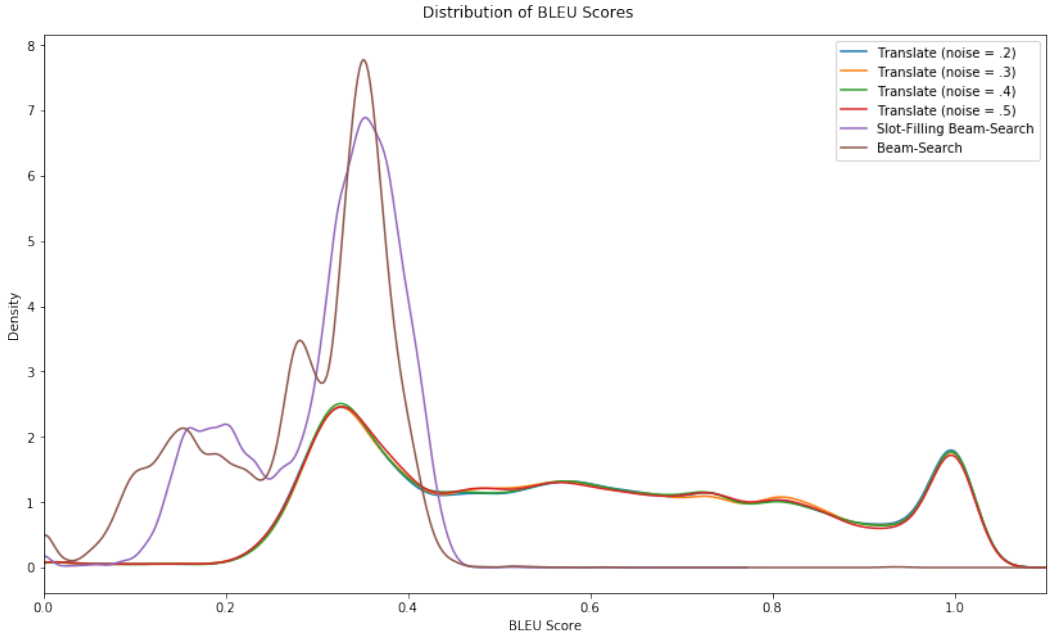

Figure 5.7.: Distribution of BLEU Scores

The relationship between between semantic and syntactical similarity is also reported in Figure 5.8. Here it becomes even more apparent, that these two measures are related, but not at all the same. Interpreting this figure it could be argued that a sentence can be changed quite a bit without losing its meaning. At some point however, any further change leads to rapid degeneration. The clearly visible clusters show once more the difference between the search and translate methods. Interestingly, the slot-filling beam-search seams to perform better than the regular beam search with higher semantic similarly and equal syntactcal similarity.

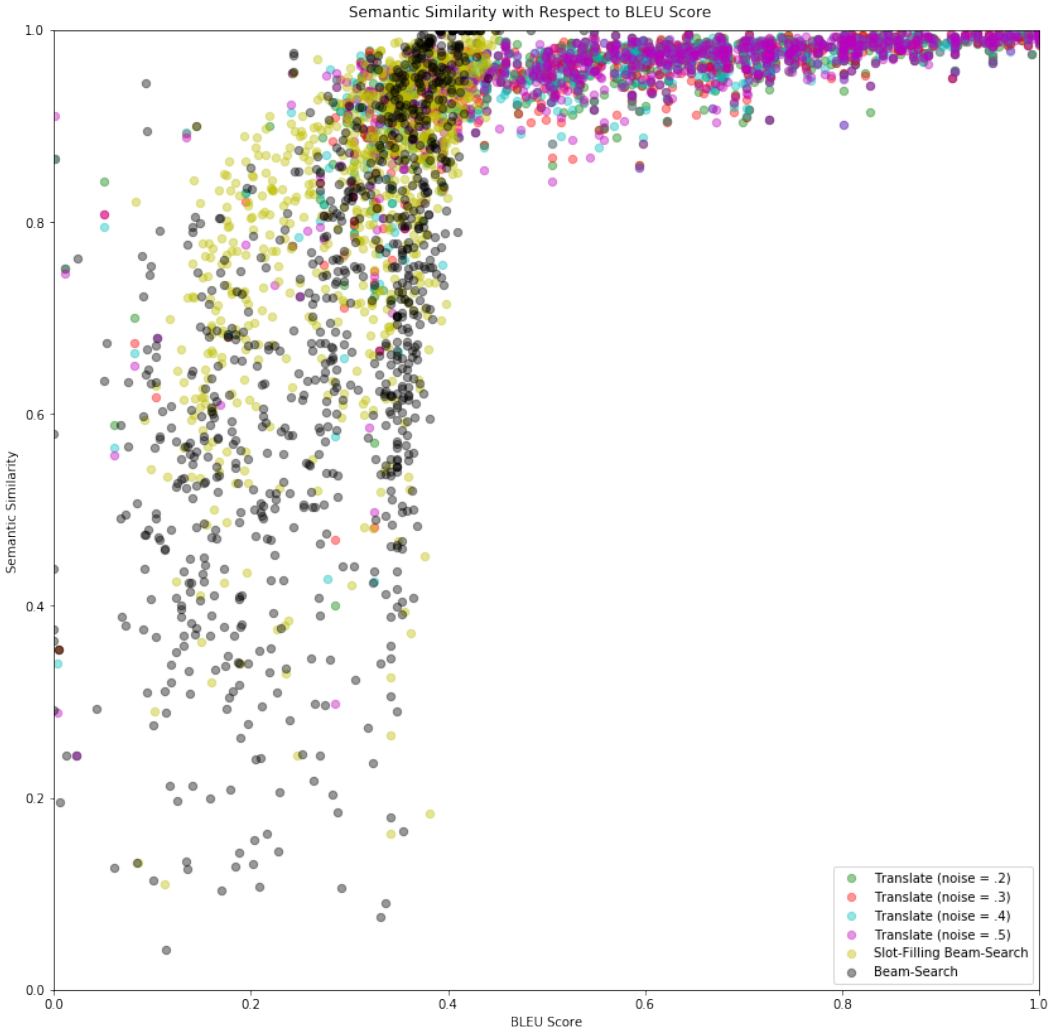

Figure 5.8.: Semantic Similarity with Respect to BLEU Score

5.2. Confirmatory Analysis

The results of the significance test performed as shown below. First, the difference in authorship scores, stated in H_1 was tested. As shown in Table 5.2 there was a significantly increase in authorship scores for all methods. As noted before, the sample size varies due to the filtering performed.

	N	M_{A0}	M_{A1}	M_d	SD_d	t_A	p_A
Translate (noise = .2)	6191	0.312	0.372	0.060	0.239	19.831	<.000
Translate (noise = .3)	6162	0.313	0.372	0.059	0.235	19.588	<.000
Translate (noise = .4)	6134	0.314	0.371	0.058	0.240	18.816	<.000
Translate (noise = .5)	6122	0.314	0.375	0.060	0.242	19.557	<.000
Slot-Filling Beam-Search	5792	0.308	0.359	0.050	0.408	9.415	<.000
Beam-Search	6065	0.306	0.484	0.178	0.460	30.205	<.000

Table 5.2.: Paired Samples t-Test Summary for Differences in Authorship Scores. N: sample size, M_{A0}: mean authorship score of source sentence, M_{A1}: mean authorship score of transformed sentence, M_d: Mean change in authorship score, SD_d: standard deviation of change in authorship score, t_A: t-Test score, p_A: p-value.

Next, the semantic similarity stated in $H_{2.1}$ was tested. Table 5.3 shows that the semantic similarity between the transformed to the original sentences is significantly higher than .75 for all methods except the regular beam search.

	N	M_S	SD_S	t_S	p_S
Translate (noise = .2)	6191	0.927	0.102	136.280	<.000
Translate (noise = .3)	6162	0.928	0.098	142.254	<.000
Translate (noise = .4)	6134	0.928	0.100	139.081	<.000
Translate (noise = .5)	6122	0.927	0.100	138.645	<.000
Slot-Filling Beam-Search	5792	0.791	0.180	17.193	<.000
Beam-Search	6065	0.652	0.237	0.000	1.000

Table 5.3.: t-Test Summary for Semantic Similarities ($H_0 : M < .75$). N: sample size, M_S: mean semantic similarity, SD_S: standard deviation of semantic similarity, t_S: t-Test score, p_S: p-value.

Last, the syntactical similarity as stated in $H_{2.2}$ was tested. The results in Table 5.4 show that the syntactical similarity between the transformed and the original sentences is significantly lower than .75 for all methods.

	N	M_B	SD_B	t_B	p_B
Translate (noise = .2)	6191	0.593	0.242	51.037	<.000
Translate (noise = .3)	6162	0.592	0.241	51.468	<.000
Translate (noise = .4)	6134	0.591	0.241	51.621	<.000
Translate (noise = .5)	6122	0.588	0.240	52.893	<.000
Slot-Filling Beam-Search	5792	0.310	0.084	396.645	<.000
Beam-Search	6065	0.282	0.100	365.471	<.000

Table 5.4.: t-Test Summary for BLEU Scores ($H_0 : M > .75$). N: sample size, M_S: mean BLEU score, SD_S: standard deviation of BLEU score, t_S: t-Test score, p_S: p-value.

In summary, all methods except one were able to prove our hypotheses. This

one being the regular beam search could not produce results that the similar enough in terms of semantic meaning.

6. Discussion

The goal of this thesis was to design an algorithm that is able to model the stylistic features of writers in order to perform transformations and translations on them. Two main methods were developed and evaluated using various settings.

The first approach employed neural machine translation techniques to perform a word-to-word translation of sentences by creating a similarity-based mapping between the word embedding spaces of authors. The same similarity measure was used to evaluate the semantic similarities between source and target sentence, which partially explains the good results regarding this measure. The syntactical similarity however showed, that the sentences changed nonetheless. While the resulting authorship score also showed a significant increase of 19%, the average absolute change of .06 is still very low. One indisputable issue with this approach is that the word order, which as mentioned before plays a major role in an author's style, stays unchanged. Additionally, the choice of anchor words posed an additional challenge, since it already leads to assumptions about similarities between authors.

The second approach took a set of predefined stylistic features and models conditional probabilities between combinations of them. Additionally, word

transition probabilities were used to improve sentence quality. The input sentence was then transformed using the first approach before feeding it as a set of words to a beam-search algorithm, evaluating the most likely word sequences. This approach has the advantage that style can be modelled independently of content and other authors. Also, this model could be used to create random sentences in the style of the author. The result showed that the new sentences were indeed closed to the target author's style with an increase of 17% (.06) and 58% (.18) for the slot-filling and regular beam-search respectively. However, the semantic similarity as well as the quality of the sentences was considerably lower than the once of the first approach. Additional restrictions and normalization would need to be added to improve the results in this regard.

7. Conclusion

The development of style transfer algorithm poses several challenges. The lack of parallel corpora makes it difficult to apply translation methods common in natural language processing. Differing and changing opinions on what is considered part of writers' styles hampers a clear separation from the content of a text. The proposed methods were able to cover some aspects of what is needed for a fully functional style transform algorithm, but many improvements need to be made.

Further work might attend to constrain the methods shown in order to improve sentence quality of the beam-search results. Alternatively, sentence restructuring algorithms may be designed to transform the word-to-word translations into a structure more representative of an author's style. Apart from improving the methods discussed in this thesis, the development of an end-to-end solution with less hyperparameters could be the superior path to go. Regardless, the applications of such algorithms will increase in the future and with it the need for good solutions.

Bibliography

Argamon, S., Koppel, M., Fine, J., & Shimoni, A. R. (2003). Gender, genre, and writing style in formal written texts. *Text-The Hague Then Amsterdam Then Berlin-*, *23*(3), 321–346.

Arora, S., Li, Y., Liang, Y., Ma, T., & Risteski, A. (2018). Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, *6*, 483–495.

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

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), 1137–1155.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Braud, C. & Søgaard, A. (2017). Is writing style predictive of scientific fraud? *arXiv preprint arXiv:1707.04095*.

Brooke, J., Hammond, A., & Hirst, G. (2015). Gutentag: an nlp-driven tool for digital humanities research in the project gutenberg corpus. In *Proceedings of the fourth workshop on computational linguistics for literature* (pp. 42–47).

Carlson, K., Riddell, A., & Rockmore, D. (2018). Evaluating prose style transfer with the bible. *Royal Society open science*, *5*(10), 171920.

Chen, B. & Cherry, C. (2014). A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the ninth workshop on statistical machine translation* (pp. 362–367).

Chen, X. [Xi], Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems* (pp. 2172–2180).

Chen, X. [Xinchi], Shi, Z., Qiu, X., & Huang, X. (2017). Adversarial multi-criteria learning for chinese word segmentation. *arXiv preprint arXiv:1704.07556*.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

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

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., & Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Crain, C. (1998). The bard's fingerprints (donald foster's controversial claim that a rather dreary funeral elegy for a seventeenth-century nobody was written by william shakespeare). *LINGUA FRANCA*, *8*(5), 28–39.

De Vel, O., Anderson, A., Corney, M., & Mohay, G. (2001). Mining e-mail content for author identification forensics. *ACM Sigmod Record*, *30*(4), 55–64.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Ficler, J. & Goldberg, Y. (2017). Controlling linguistic style aspects in neural language generation. *arXiv preprint arXiv:1707.02633*.

Forney, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, *61*(3), 268–278.

Fu, Z., Tan, X., Peng, N., Zhao, D., & Yan, R. (2018). Style transfer in text: exploration and evaluation. In *Thirty-second aaai conference on artificial intelligence*.

Ganin, Y. & Lempitsky, V. (2014). Unsupervised domain adaptation by back-propagation. *arXiv preprint arXiv:1409.7495*.

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2414–2423).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Gower, J. C., Dijksterhuis, G. B. et al. (2004). *Procrustes problems*. Oxford University Press on Demand.

Haykin, S. (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

Haykin, S. (2009). *Neural networks and learning machines/simon haykin.* New York: Prentice Hall.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *CoRR, abs/1512.03385.* arXiv: 1512.03385. Retrieved from http://arxiv.org/abs/1512.03385

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., & Salesin, D. H. (2001). Image analogies. In *Proceedings of the 28th annual conference on computer graphics and interactive techniques* (pp. 327–340). ACM.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735–1780.

Honnibal, M. & Montani, I. (2017). *spaCy 2: natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.* To appear.

Hope, J. (1994). *The authorship of shakespeare's plays: a socio-linguistic study.* Cambridge University Press.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., & Xing, E. P. (2017). Toward controlled generation of text. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 1587–1596). JMLR. org.

Hu, Z., Yang, Z., Salakhutdinov, R. R., Qin, L., Liang, X., Dong, H., & Xing, E. P. (2018). Deep generative models with learnable knowledge constraints. In *Advances in neural information processing systems* (pp. 10501–10512).

Jhamtani, H., Gangal, V., Hovy, E., & Nyberg, E. (2017). Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161.*

Juola, P. et al. (2008). Authorship attribution. *Foundations and Trends® in Information Retrieval, 1*(3), 233–334.

Jurafsky, D. & Martin, J. H. (2009). *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, N.J.: Pearson Prentice Hall.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *CoRR*, *abs/1408.5882*. arXiv: 1408.5882. Retrieved from http://arxiv.org/abs/1408.5882

Kingma, D. P. [Diederik P] & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kingma, D. P. [Durk P], Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems* (pp. 3581–3589).

Koppel, M., Schler, J., & Argamon, S. (2009). Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, *60*(1), 9–26.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., ... Socher, R. (2016). Ask me anything: dynamic memory networks for natural language processing. In *International conference on machine learning* (pp. 1378–1387).

Lample, G., Conneau, A., Denoyer, L., & Ranzato, M. (2017). Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Lample, G., Subramanian, S., Smith, E., Denoyer, L., Boureau, Y.-L., et al. (2018). Multiple-attribute text rewriting.

LeCun, Y. et al. (1989). Generalization and network design strategies. In *Connectionism in perspective* (Vol. 19). Citeseer.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

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

Malinowski, M., Rohrbach, M., & Fritz, M. (2015). Ask your neurons: a neural-based approach to answering questions about images. In *Proceedings of the ieee international conference on computer vision* (pp. 1–9).

Manning, C. D. & Schütze, H. (1999). Foundations of statistical natural language processing.

Marcus, M., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: the penn treebank.

Markov, A. A. (1954). The theory of algorithms. *Trudy Matematicheskogo Instituta Imeni VA Steklova*, *42*, 3–375.

McClelland, J. L., Rumelhart, D. E., Group, P. R., et al. (1987). *Parallel distributed processing*. MIT press Cambridge, MA:

McMenamin, G. R. (2002). *Forensic linguistics: advances in forensic stylistics*. CRC press.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Mo, S., Cho, M., & Shin, J. (2018). Instagan: instance-aware image-to-image translation. *arXiv preprint arXiv:1812.10889*.

Mueller, J., Gifford, D., & Jaakkola, T. (2017). Sequence to better sequence: continuous revision of combinatorial structures. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 2536–2544). JMLR. org.

Project Gutenberg. (2019). Retrieved from http://www.gutenberg.org/

Pan, S. J. & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345–1359.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (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.

Peng, N., Ghazvininejad, M., May, J., & Knight, K. (2018). Towards controllable story generation. In *Proceedings of the first workshop on storytelling* (pp. 43–49).

Pennington, J., Socher, R., & Manning, C. (2014). Glove: global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Petrov, S., Das, D., & McDonald, R. (2011). A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., & Black, A. W. (2018). Style transfer through back-translation. *arXiv preprint arXiv:1804.09000*.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65–386.

Roser, M. & Ortiz-Ospina, E. (2019). Literacy. *Our World in Data*. https://ourworldindata.org/litera

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science.

Schandry, R. (2007). *Biologische psychologie*.

Schmied, J. (2019). The english / german translation corpus. TU Chemnitz. Retrieved from https://www.tu-chemnitz.de/phil/english/sections/linguist/real/independent/transcorpus/index.htm

Schoenbaum, S. (1966). *Internal evidence and elizabethan dramatic authorship: an essay in literary history and method*. Northwestern University Press.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, *27*(3), 379–423.

Shen, T., Lei, T., Barzilay, R., & Jaakkola, T. (2017). Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems* (pp. 6830–6841).

Sundermeyer, M., Alkhouli, T., Wuebker, J., & Ney, H. (2014). Translation modeling with bidirectional recurrent neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 14–25).

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

Tenenbaum, J. B. & Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural computation*, *12*(6), 1247–1283.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In *Lrec* (Vol. 2012, pp. 2214–2218).

Tweedie, F. J. & Baayen, R. H. (1998). How variable may a constant be? measures of lexical richness in perspective. *Computers and the Humanities*, *32*(5), 323–352.

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

Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks. In *Advances in neural information processing systems* (pp. 2692–2700).

Viterbi, A. J. (2010). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In *The foundations of the digital wireless world: selected works of aj viterbi* (pp. 41–50). World Scientific.

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

Xu, W., Ritter, A., Dolan, B., Grishman, R., & Cherry, C. (2012). Paraphrasing for style. In *Proceedings of coling 2012* (pp. 2899–2914).

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., & Berg-Kirkpatrick, T. (2018). Unsupervised text style transfer using language models as discriminators. In *Advances in neural information processing systems* (pp. 7287–7298).

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.

Yin, Z. & Shen, Y. (2018). On the dimensionality of word embedding. In *Advances in neural information processing systems* (pp. 887–898).

Zeng, D., Liu, K., Lai, S., Zhou, G., Zhao, J., et al. (2014). Relation classification via convolutional deep neural network.

Zhang, Z. & Luo, L. (2018). Hate speech detection: a solved problem? the challenging case of long tail on twitter. *Semantic Web*, (Preprint), 1–21.

Zheng, R., Li, J., Chen, H., & Huang, Z. (2006). A framework for authorship identification of online messages: writing-style features and classification techniques. *Journal of the American society for information science and technology*, *57*(3), 378–393.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer vision (iccv), 2017 ieee international conference on*.

Ziemski, M., Junczys-Dowmunt, M., & Pouliquen, B. (2016). The united nations parallel corpus v1. 0. In *Proceedings of the tenth international conference on language resources and evaluation (lrec 2016)* (pp. 3530–3534).

Zipf, G. K. (1949). Human behavior and the principle of least effort.

Appendix

Appendix A.

Evaluation Results per Author

A.1. Gilbert Keith Chesterton

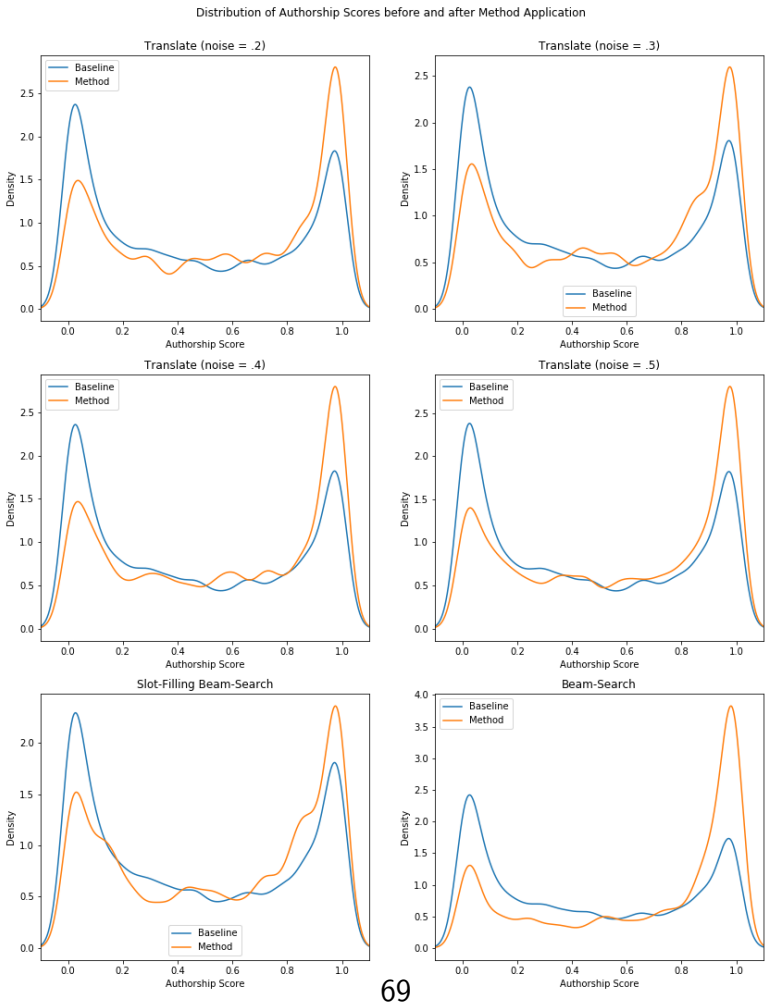

Figure A.1.: Chesterton: Distribution of Authorship Scores before and after Method Application

Appendix A. Evaluation Results per Author

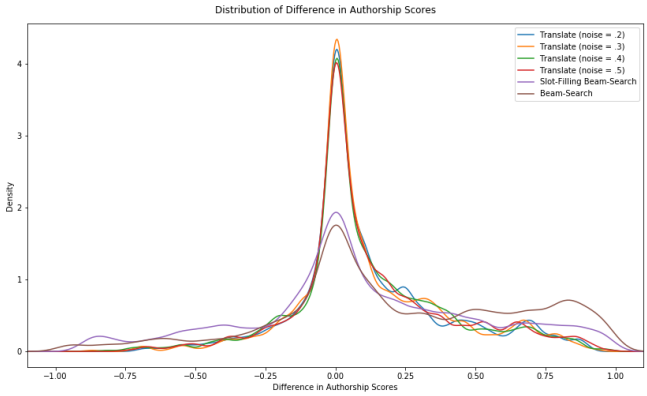

Figure A.2.: Chesterton: Distribution of Difference in Authorship Scores

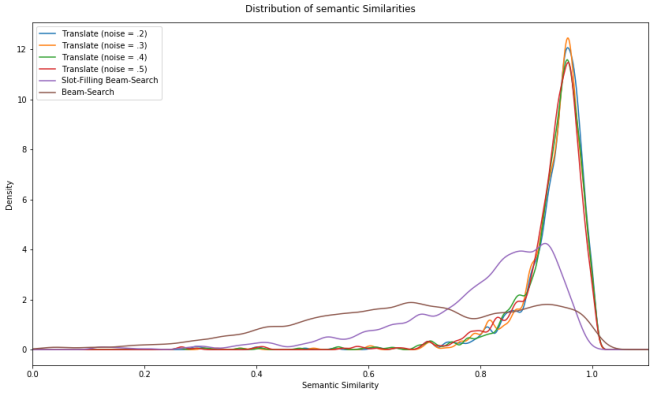

Figure A.3.: Chesterton: Distribution of semantic Similarities

Appendix A. Evaluation Results per Author

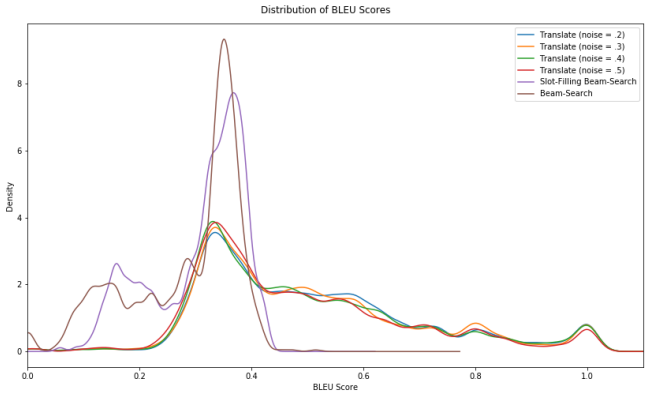

Figure A.4.: Chesterton: Distribution of BLEU Scores

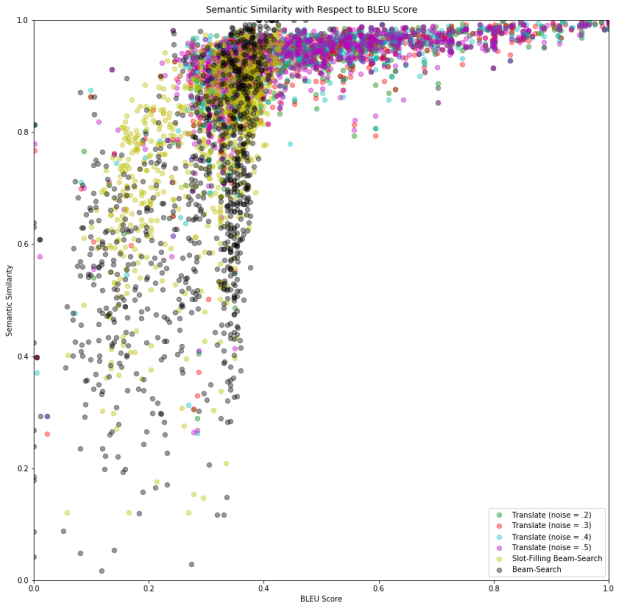

Figure A.5.: Chesterton: Semantic Similarity with Respect to BLEU Score

Appendix A. Evaluation Results per Author

	N	M_{A0}	M_{A1}	M_d	SD_d	t_A	p_A
Translate (noise = .2)	912	0.457	0.571	0.114	0.271	12.734	.000
Translate (noise = .3)	911	0.455	0.569	0.114	0.267	12.859	.000
Translate (noise = .4)	912	0.458	0.570	0.112	0.277	12.276	.000
Translate (noise = .5)	908	0.458	0.577	0.119	0.281	12.789	.000
Slot-Filling Beam-Search	843	0.460	0.556	0.096	0.412	6.755	.000
Beam-Search	859	0.451	0.662	0.212	0.438	14.148	.000

Table A.1.: Chesterton: Paired Samples t-Test Summary for Differences in Authorship Scores

	N	M_S	SD_S	t_S	p_S
Translate (noise = .2)	912	0.929	0.073	73.972	.000
Translate (noise = .3)	911	0.927	0.073	73.368	.000
Translate (noise = .4)	912	0.925	0.076	69.084	.000
Translate (noise = .5)	908	0.922	0.077	67.100	.000
Slot-Filling Beam-Search	843	0.797	0.148	9.247	.000
Beam-Search	859	0.676	0.211	0.000	1.000

Table A.2.: Chesterton: t-Test Summary for Semantic Similarities ($H_0 : M < .75$)

	N	M_B	SD_B	t_B	p_B
Translate (noise = .2)	912	0.506	0.196	37.702	.000
Translate (noise = .3)	911	0.502	0.196	38.330	.000
Translate (noise = .4)	912	0.497	0.196	38.832	.000
Translate (noise = .5)	908	0.484	0.190	42.175	.000
Slot-Filling Beam-Search	843	0.306	0.084	154.290	.000
Beam-Search	859	0.283	0.101	136.166	.000

Table A.3.: Chesterton: t-Test Summary for BLEU-Scores ($H_0 : M > .75$)

A.2. Winston Churchill

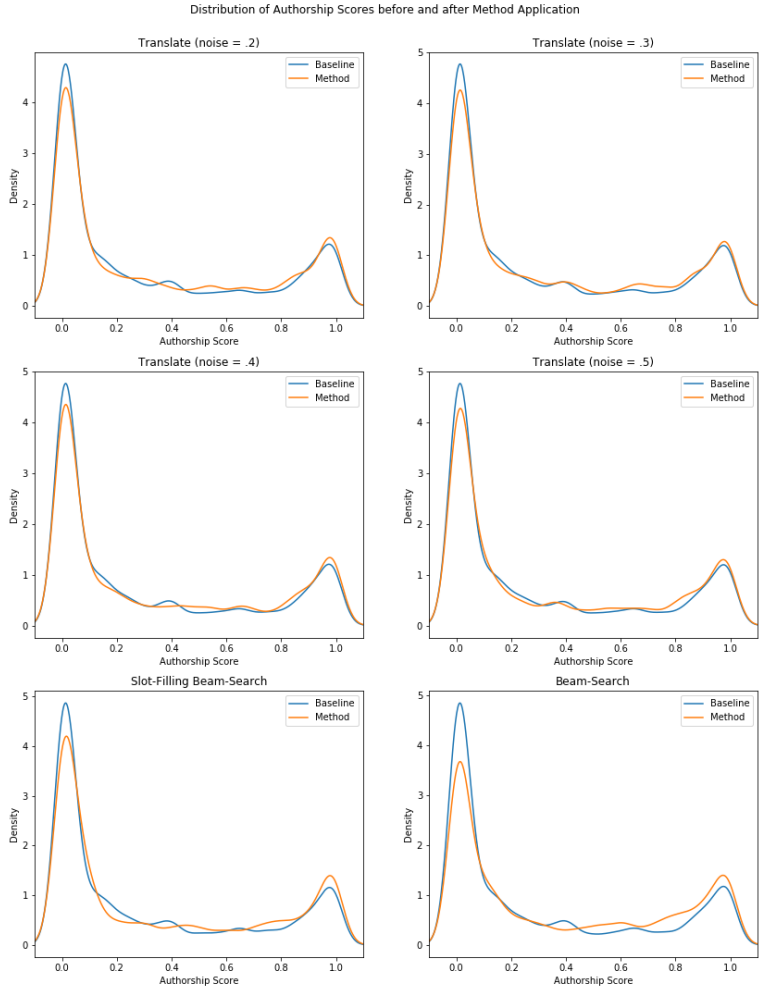

Figure A.6.: Churchill: Distribution of Authorship Scores before and after Method Application

Appendix A. Evaluation Results per Author

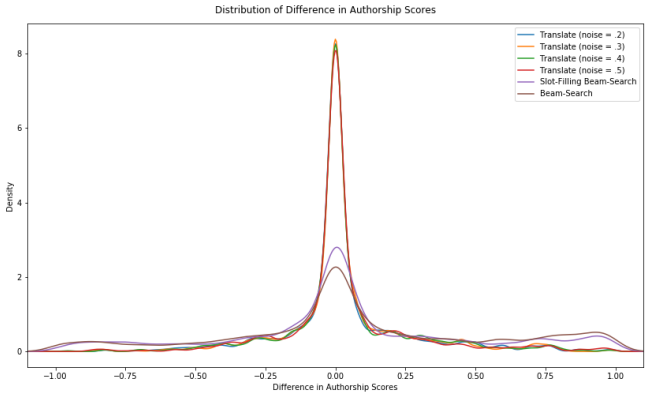

Figure A.7.: Churchill: Distribution of Difference in Authorship Scores

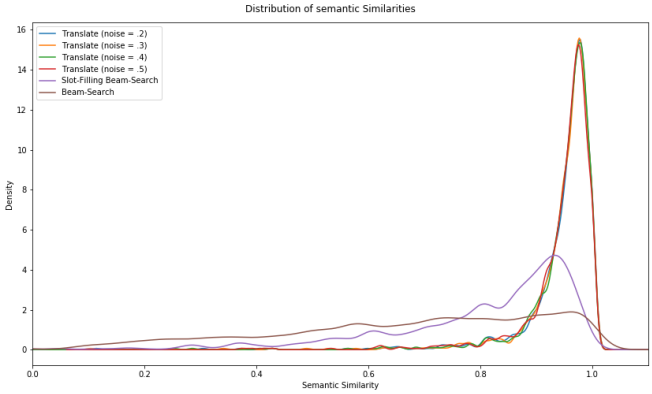

Figure A.8.: Churchill: Distribution of semantic Similarities

Appendix A. Evaluation Results per Author

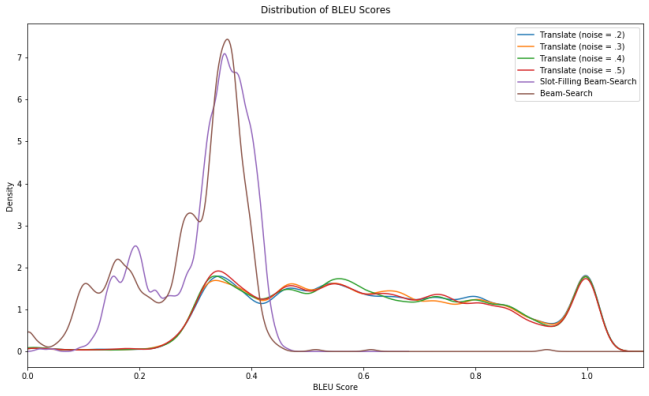

Figure A.9.: Churchill: Distribution of BLEU Scores

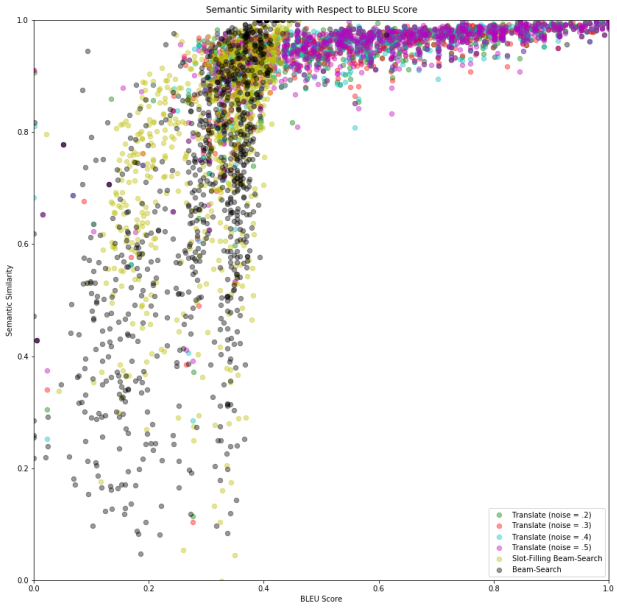

Figure A.10.: Churchill: Semantic Similarity with Respect to BLEU Score

Appendix A. Evaluation Results per Author

	N	M_{A0}	M_{A1}	M_d	SD_d	t_A	p_A
Translate (noise = .2)	899	0.302	0.332	0.031	0.222	4.159	.000
Translate (noise = .3)	899	0.299	0.330	0.031	0.216	4.316	.000
Translate (noise = .4)	898	0.301	0.331	0.030	0.221	4.109	.000
Translate (noise = .5)	899	0.300	0.330	0.030	0.229	3.978	.000
Slot-Filling Beam-Search	809	0.294	0.337	0.043	0.438	2.791	.003
Beam-Search	844	0.295	0.372	0.077	0.475	4.689	.000

Table A.4.: Churchill: Paired Samples t-Test Summary for Differences in Authorship Scores

	N	M_S	SD_S	t_S	p_S
Translate (noise = .2)	899	0.948	0.070	84.479	.000
Translate (noise = .3)	899	0.948	0.071	83.636	.000
Translate (noise = .4)	898	0.948	0.068	87.388	.000
Translate (noise = .5)	899	0.946	0.068	86.935	.000
Slot-Filling Beam-Search	809	0.803	0.172	0.000	1.000
Beam-Search	844	0.663	0.255	0.000	1.000

Table A.5.: Churchill: t-Test Summary for Semantic Similarities ($H_0 : M < .75$)

	N	M_B	SD_B	t_B	p_B
Translate (noise = .2)	899	0.625	0.225	16.668	.000
Translate (noise = .3)	899	0.622	0.224	17.162	.000
Translate (noise = .4)	898	0.620	0.224	17.313	.000
Translate (noise = .5)	899	0.616	0.224	17.992	.000
Slot-Filling Beam-Search	809	0.318	0.082	150.064	.000
Beam-Search	844	0.287	0.102	131.595	.000

Table A.6.: Churchill: t-Test Summary for BLEU-Scores ($H_0 : M > .75$)

A.3. Charles Darwin

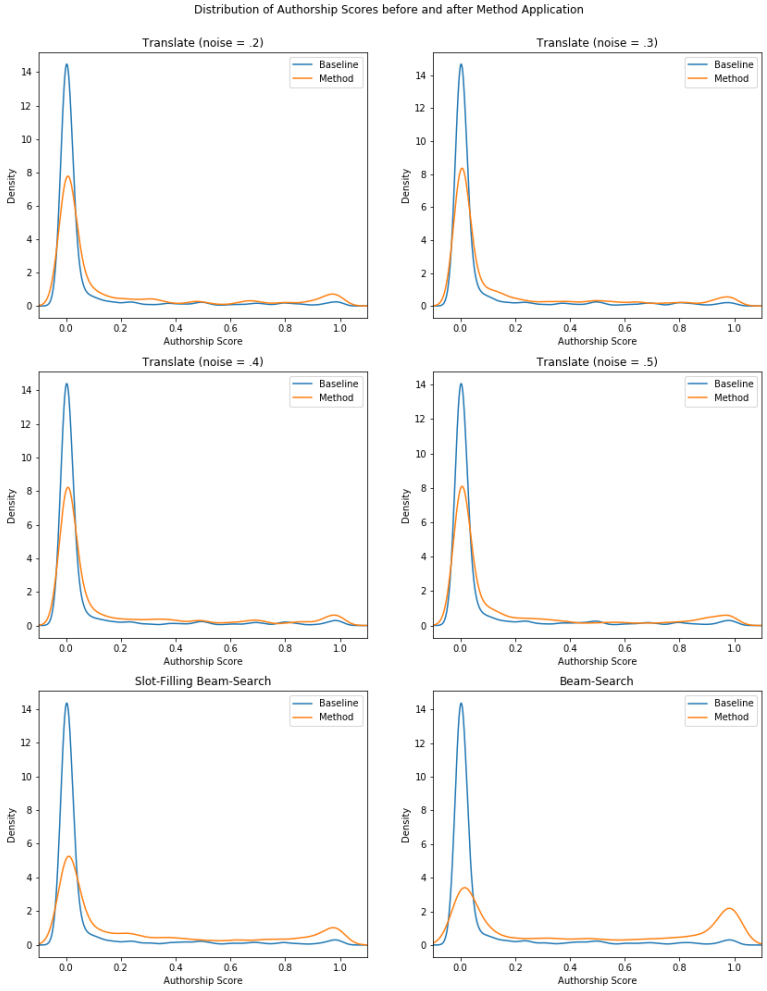

Figure A.11.: Darwin: Distribution of Authorship Scores before and after Method Application

Appendix A. Evaluation Results per Author

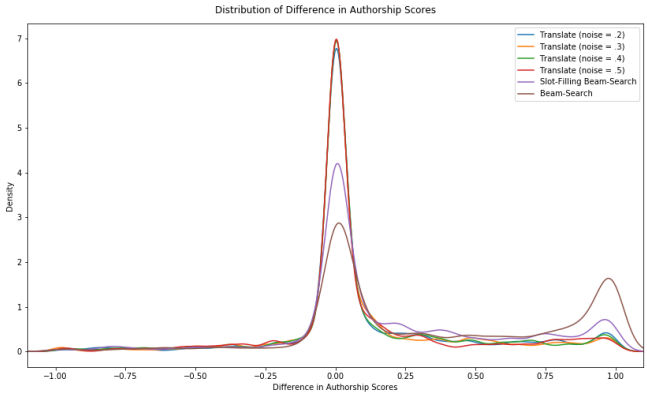

Figure A.12.: Darwin: Distribution of Difference in Authorship Scores

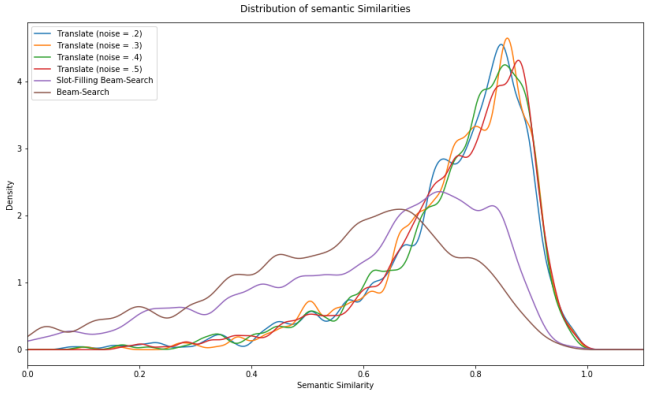

Figure A.13.: Darwin: Distribution of semantic Similarities

Appendix A. Evaluation Results per Author

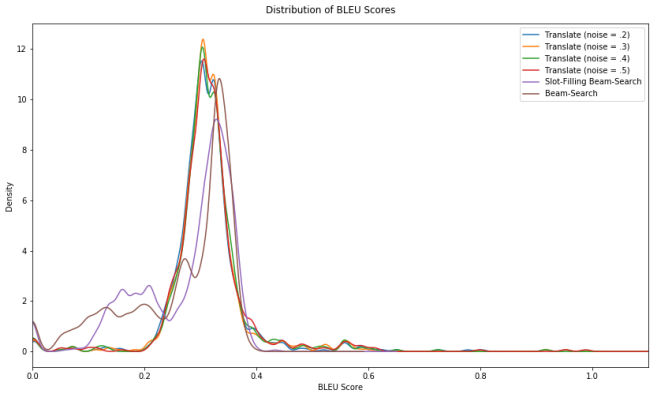

Figure A.14.: Darwin: Distribution of BLEU Scores

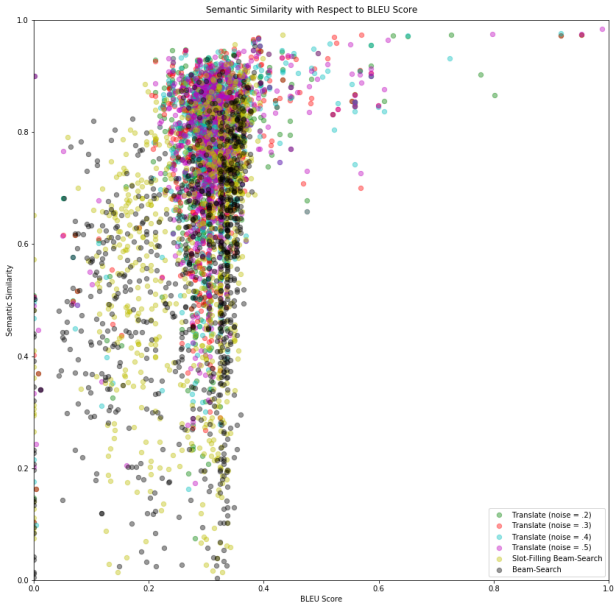

Figure A.15.: Darwin: Semantic Similarity with Respect to BLEU Score

Appendix A. Evaluation Results per Author

	N	M_{A0}	M_{A1}	M_d	SD_d	t_A	p_A
Translate (noise = .2)	786	0.080	0.180	0.100	0.321	8.704	.000
Translate (noise = .3)	760	0.079	0.165	0.086	0.307	7.690	.000
Translate (noise = .4)	740	0.080	0.170	0.089	0.317	7.665	.000
Translate (noise = .5)	728	0.084	0.170	0.086	0.315	7.379	.000
Slot-Filling Beam-Search	826	0.081	0.271	0.190	0.388	14.059	.000
Beam-Search	823	0.081	0.426	0.345	0.455	21.708	.000

Table A.7.: Darwin: Paired Samples t-Test Summary for Differences in Authorship Scores

	N	M_S	SD_S	t_S	p_S
Translate (noise = .2)	786	0.759	0.141	1.830	.034
Translate (noise = .3)	760	0.766	0.131	3.272	.001
Translate (noise = .4)	740	0.763	0.137	2.581	.005
Translate (noise = .5)	728	0.763	0.137	2.609	.005
Slot-Filling Beam-Search	826	0.603	0.218	0.000	1.000
Beam-Search	823	0.537	0.223	0.000	1.000

Table A.8.: Darwin: t-Test Summary for Semantic Similarities ($H_0 : M < .75$)

	N	M_B	SD_B	t_B	p_B
Translate (noise = .2)	786	0.313	0.077	159.822	.000
Translate (noise = .3)	760	0.313	0.071	168.504	.000
Translate (noise = .4)	740	0.314	0.073	162.185	.000
Translate (noise = .5)	728	0.316	0.080	147.129	.000
Slot-Filling Beam-Search	826	0.277	0.085	159.765	.000
Beam-Search	823	0.265	0.096	144.493	.000

Table A.9.: Darwin: t-Test Summary for BLEU-Scores ($H_0 : M > .75$)

A.4. Charles Dickens

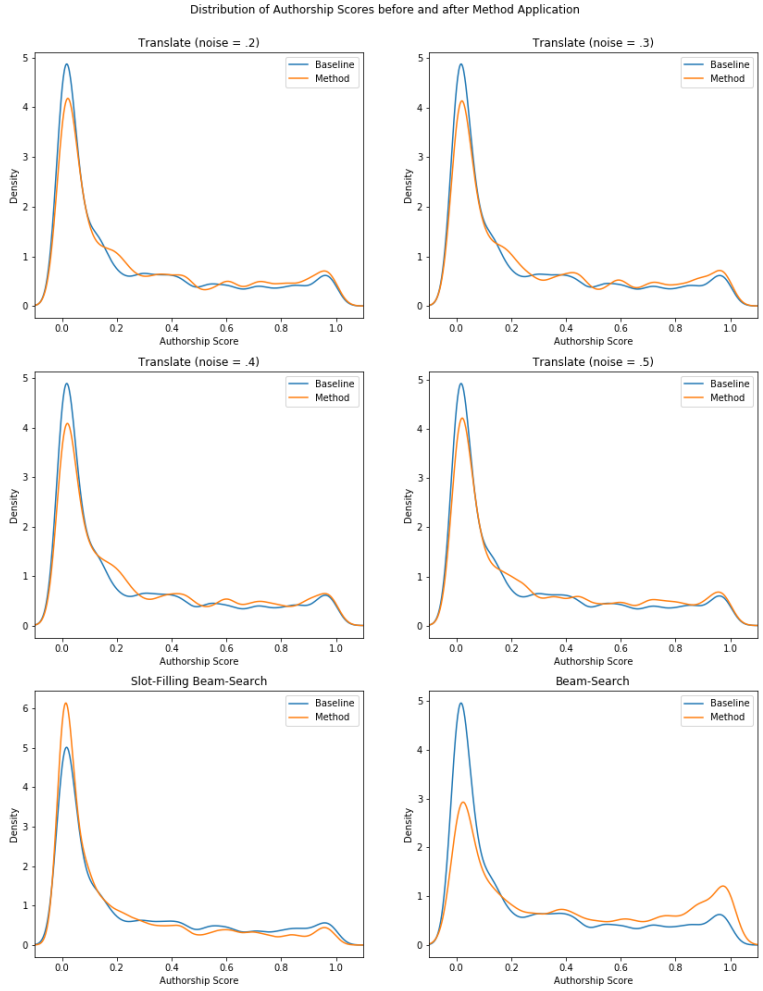

Figure A.16.: Dickens: Distribution of Authorship Scores before and after Method Application

Appendix A. Evaluation Results per Author

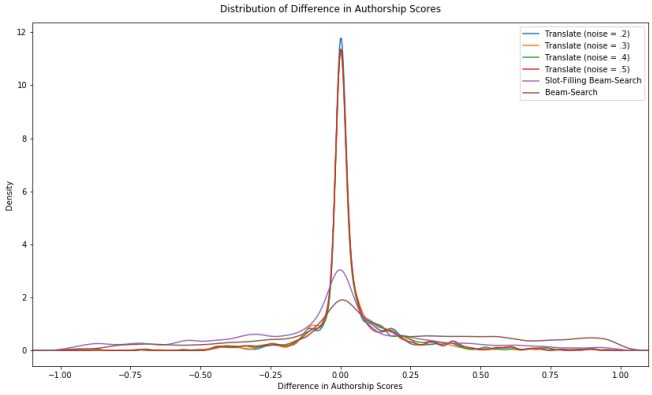

Figure A.17.: Dickens: Distribution of Difference in Authorship Scores

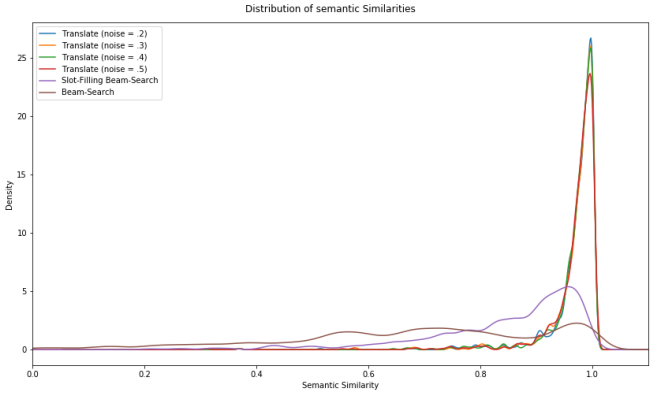

Figure A.18.: Dickens: Distribution of semantic Similarities

Appendix A. Evaluation Results per Author

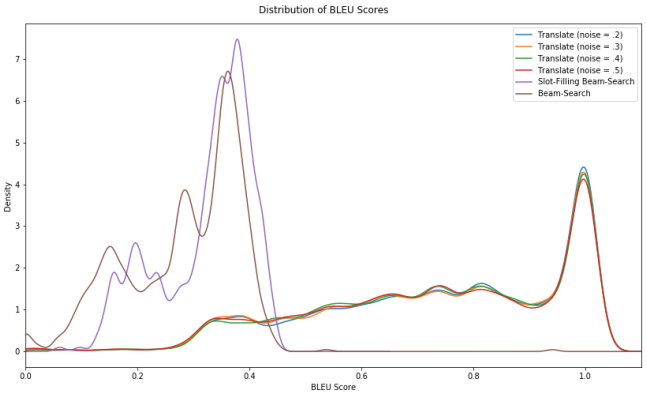

Figure A.19.: Dickens: Distribution of BLEU Scores

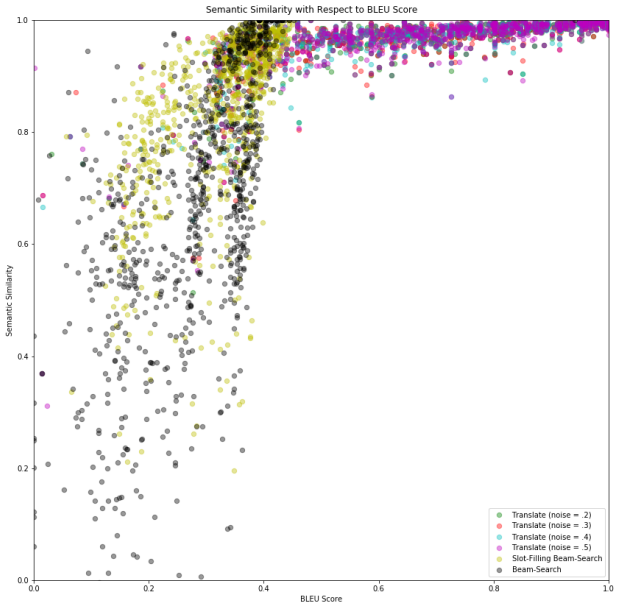

Figure A.20.: Dickens: Semantic Similarity with Respect to BLEU Score

Appendix A. Evaluation Results per Author

	N	M_{A0}	M_{A1}	M_d	SD_d	t_A	p_A
Translate (noise = .2)	894	0.260	0.294	0.033	0.164	6.030	.000
Translate (noise = .3)	895	0.260	0.295	0.036	0.164	6.497	.000
Translate (noise = .4)	899	0.259	0.290	0.031	0.167	5.592	.000
Translate (noise = .5)	902	0.258	0.292	0.034	0.168	6.038	.000
Slot-Filling Beam-Search	832	0.255	0.203	0.052	0.365	4.099	.000
Beam-Search	866	0.259	0.394	0.136	0.431	9.256	.000

Table A.10.: Dickens: Paired Samples t-Test Summary for Differences in Authorship Scores

	N	M_S	SD_S	t_S	p_S
Translate (noise = .2)	894	0.971	0.048	138.373	.000
Translate (noise = .3)	895	0.970	0.050	133.033	.000
Translate (noise = .4)	899	0.971	0.048	138.362	.000
Translate (noise = .5)	902	0.969	0.055	119.619	.000
Slot-Filling Beam-Search	832	0.844	0.147	18.462	.000
Beam-Search	866	0.678	0.241	0.000	1.000

Table A.11.: Dickens: t-Test Summary for Semantic Similarities ($H_0 : M < .75$)

	N	M_B	SD_B	t_B	p_B
Translate (noise = .2)	894	0.748	0.221	0.324	.373
Translate (noise = .3)	895	0.743	0.222	0.994	.160
Translate (noise = .4)	899	0.743	0.218	0.950	.171
Translate (noise = .5)	902	0.737	0.224	1.743	.041
Slot-Filling Beam-Search	832	0.323	0.082	149.664	.000
Beam-Search	866	0.284	0.103	133.126	.000

Table A.12.: Dickens: t-Test Summary for BLEU-Scores ($H_0 : M > .75$)

A.5. Mark Twain

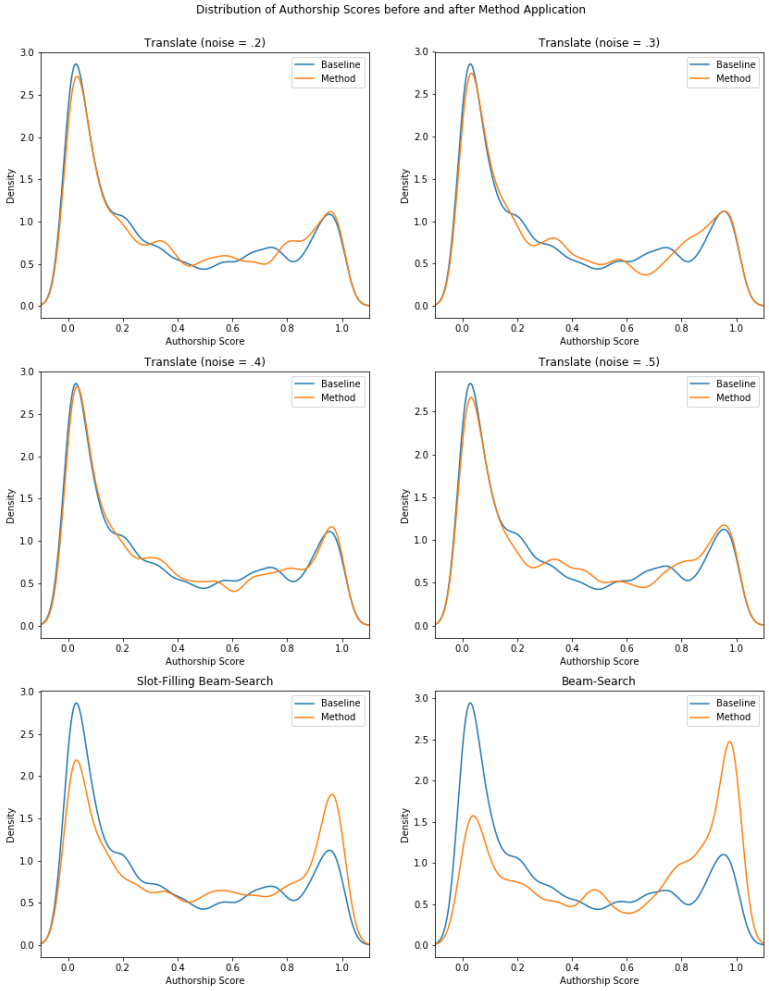

Figure A.21.: Twain: Distribution of Authorship Scores before and after Method Application

Appendix A. Evaluation Results per Author

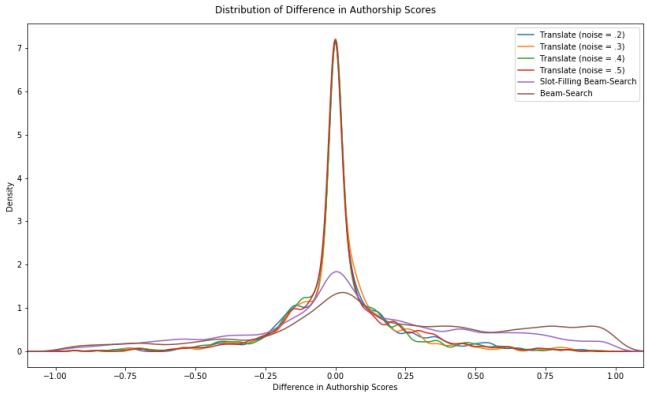

Figure A.22.: Twain: Distribution of Difference in Authorship Scores

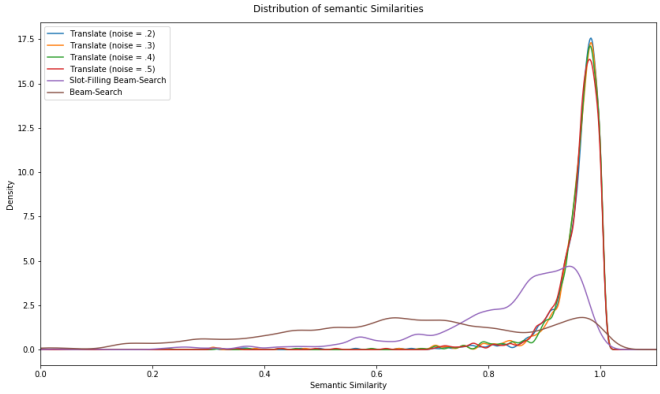

Figure A.23.: Twain: Distribution of semantic Similarities

Appendix A. Evaluation Results per Author

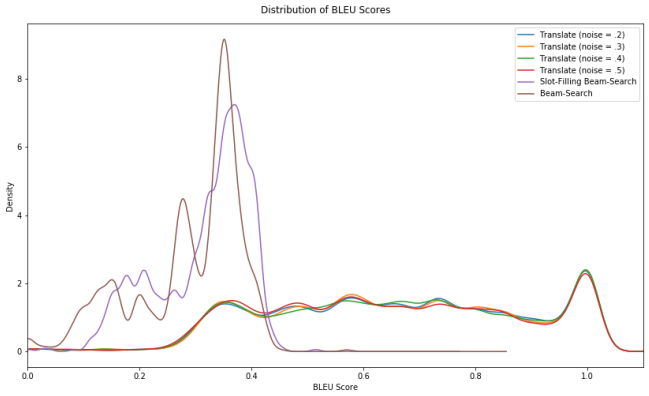

Figure A.24.: Twain: Distribution of BLEU Scores

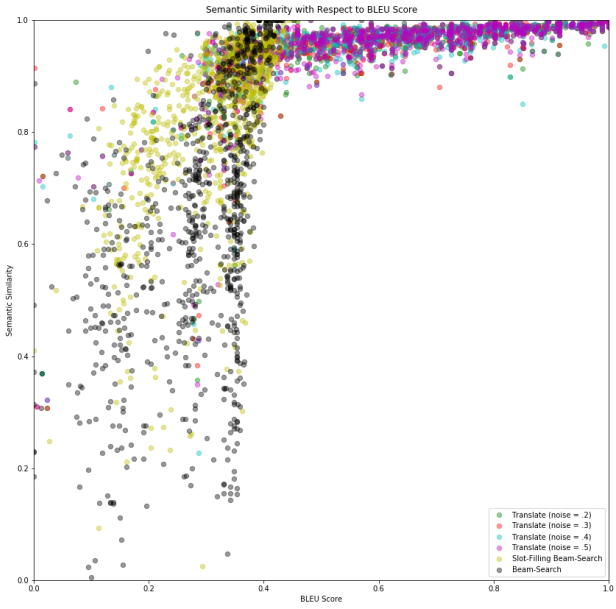

Figure A.25.: Twain: Semantic Similarity with Respect to BLEU Score

Appendix A. Evaluation Results per Author

	N	M_{A0}	M_{A1}	M_d	SD_d	t_A	p_A
Translate (noise = .2)	899	0.379	0.395	0.016	0.205	2.294	.011
Translate (noise = .3)	903	0.382	0.391	0.009	0.204	1.334	.091
Translate (noise = .4)	898	0.381	0.386	0.005	0.202	0.761	.223
Translate (noise = .5)	898	0.383	0.401	0.018	0.200	2.681	.004
Slot-Filling Beam-Search	837	0.382	0.465	0.083	0.408	5.919	.000
Beam-Search	884	0.375	0.561	0.186	0.459	12.076	.000

Table A.13.: Twain: Paired Samples t-Test Summary for Differences in Authorship Scores

	N	M_S	SD_S	t_S	p_S
Translate (noise = .2)	899	0.958	0.062	100.499	.000
Translate (noise = .3)	903	0.957	0.064	97.911	.000
Translate (noise = .4)	898	0.957	0.064	96.422	.000
Translate (noise = .5)	898	0.956	0.065	95.405	.000
Slot-Filling Beam-Search	837	0.828	0.152	14.930	.000
Beam-Search	884	0.645	0.243	0.000	1.000

Table A.14.: Twain: t-Test Summary for Semantic Similarities ($H_0 : M < .75$)

	N	M_B	SD_B	t_B	p_B
Translate (noise = .2)	899	0.661	0.227	11.719	.000
Translate (noise = .3)	903	0.660	0.227	11.911	.000
Translate (noise = .4)	898	0.660	0.228	11.822	.000
Translate (noise = .5)	898	0.652	0.228	12.909	.000
Slot-Filling Beam-Search	837	0.315	0.083	151.147	.000
Beam-Search	884	0.289	0.096	142.107	.000

Table A.15.: Twain: t-Test Summary for BLEU-Scores ($H_0 : M > .75$)

A.6. Herbert George Wells

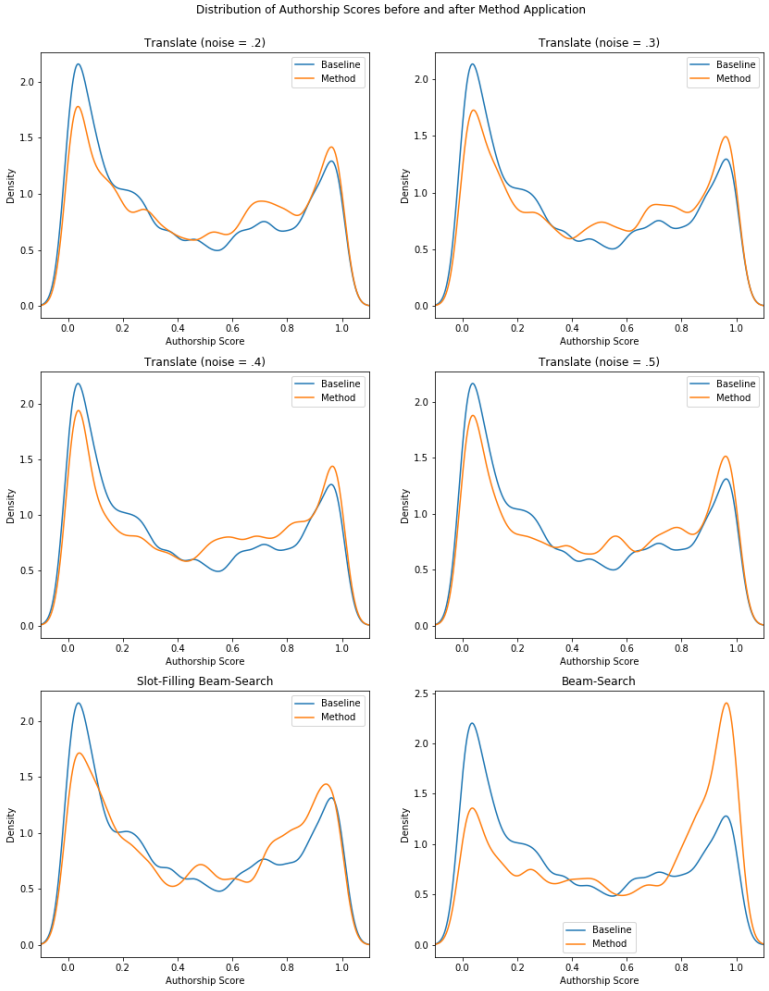

Figure A.26.: Wells: Distribution of Authorship Scores before and after Method Application

Appendix A. Evaluation Results per Author

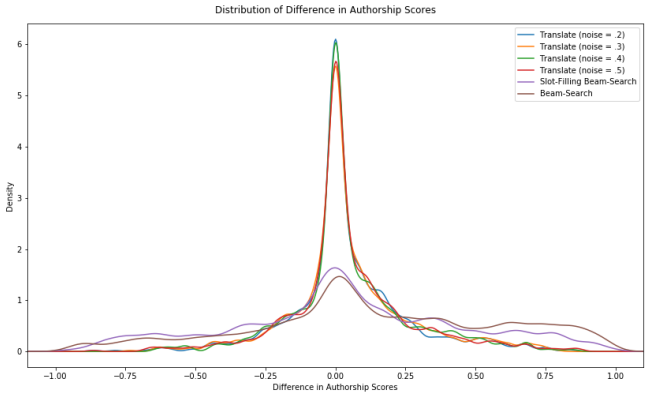

Figure A.27.: Wells: Distribution of Difference in Authorship Scores

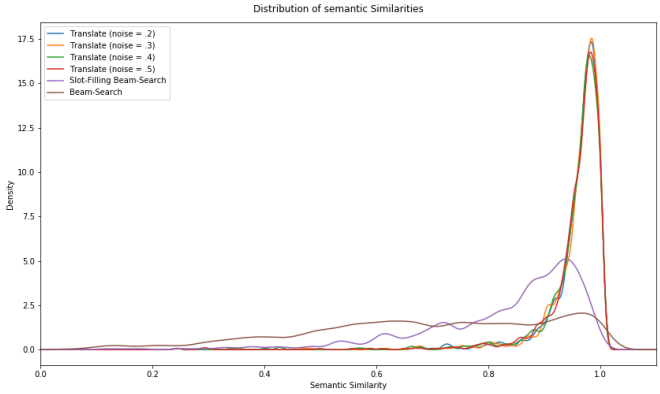

Figure A.28.: Wells: Distribution of semantic Similarities

Appendix A. Evaluation Results per Author

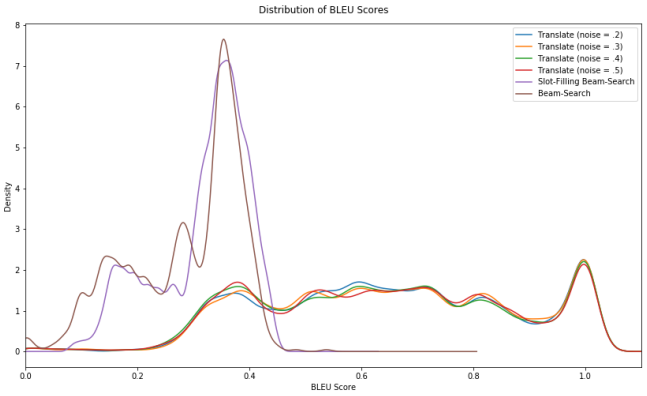

Figure A.29.: Wells: Distribution of BLEU Scores

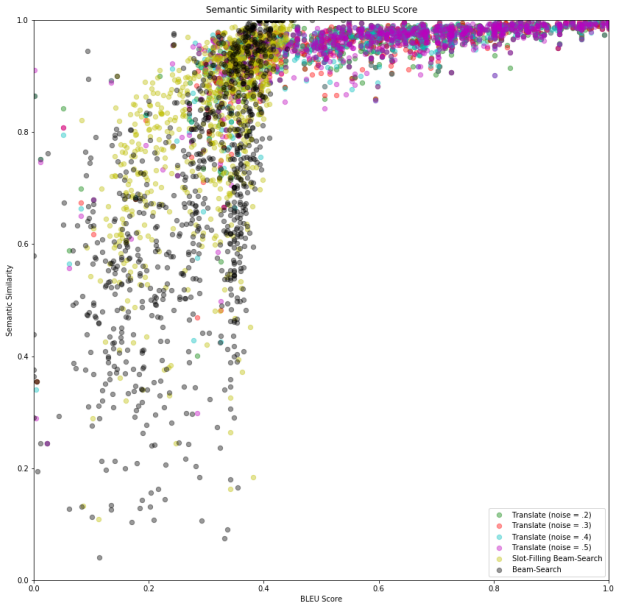

Figure A.30.: Wells: Semantic Similarity with Respect to BLEU Score

	N	M_{A0}	M_{A1}	M_d	SD_d	t_A	p_A
Translate (noise = .2)	893	0.434	0.477	0.043	0.209	6.183	.000
Translate (noise = .3)	896	0.436	0.483	0.047	0.215	6.535	.000
Translate (noise = .4)	890	0.432	0.480	0.048	0.212	6.749	.000
Translate (noise = .5)	891	0.434	0.479	0.045	0.219	6.168	.000
Slot-Filling Beam-Search	823	0.439	0.479	0.040	0.415	2.762	.003
Beam-Search	892	0.431	0.566	0.136	0.448	9.043	.000

Table A.16.: Wells: Paired Samples t-Test Summary for Differences in Authorship Scores

	N	M_S	SD_S	t_S	p_S
Translate (noise = .2)	893	0.957	0.061	100.851	.000
Translate (noise = .3)	896	0.957	0.061	101.941	.000
Translate (noise = .4)	890	0.955	0.065	93.821	.000
Translate (noise = .5)	891	0.955	0.066	93.097	.000
Slot-Filling Beam-Search	823	0.832	0.144	16.211	.000
Beam-Search	892	0.687	0.226	0.000	1.000

Table A.17.: Wells: t-Test Summary for Semantic Similarities ($H_0 : M < .75$)

	N	M_B	SD_B	t_B	p_B
Translate (noise = .2)	893	0.652	0.223	13.088	.000
Translate (noise = .3)	896	0.654	0.224	12.807	.000
Translate (noise = .4)	890	0.645	0.226	13.848	.000
Translate (noise = .5)	891	0.647	0.225	13.698	.000
Slot-Filling Beam-Search	823	0.317	0.082	151.249	.000
Beam-Search	892	0.286	0.100	138.663	.000

Table A.18.: Wells: t-Test Summary for BLEU-Scores ($H_0 : M > .75$)

A.7. Jules Verne

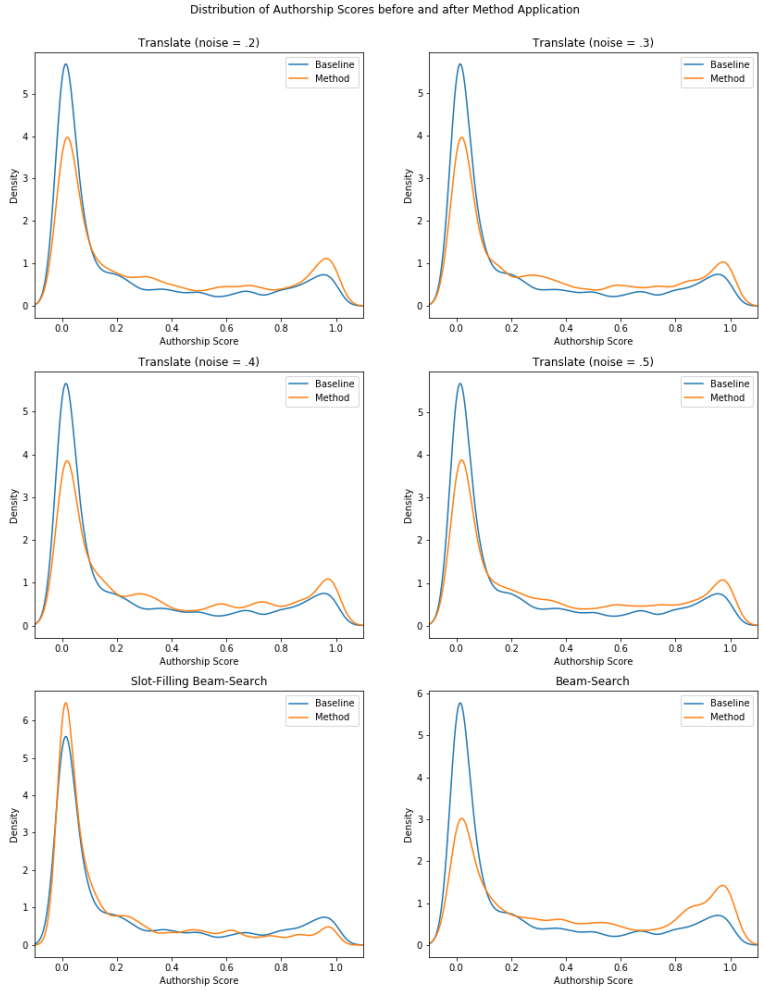

Figure A.31.: Verne: Distribution of Authorship Scores before and after Method Application

Appendix A. Evaluation Results per Author

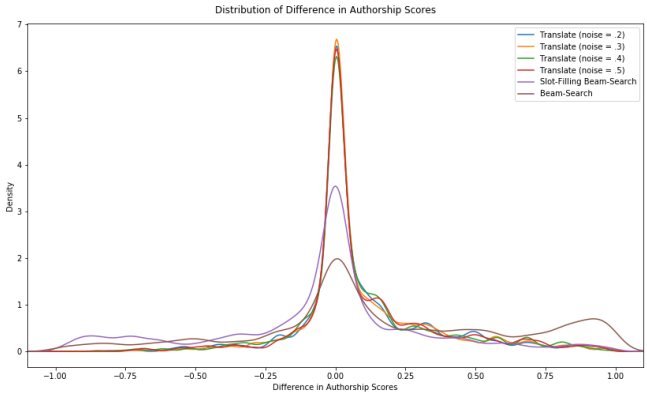

Figure A.32.: Verne: Distribution of Difference in Authorship Scores

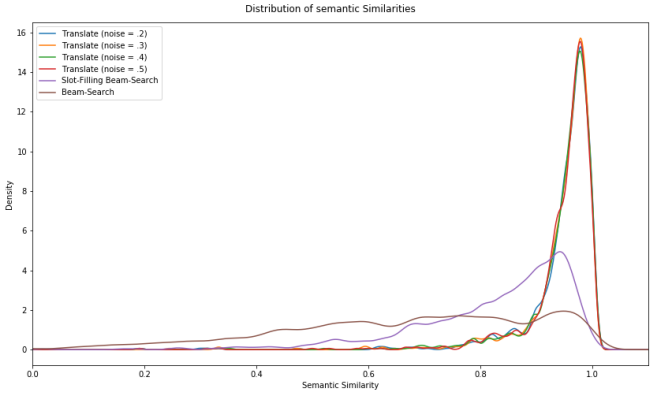

Figure A.33.: Verne: Distribution of semantic Similarities

Appendix A. Evaluation Results per Author

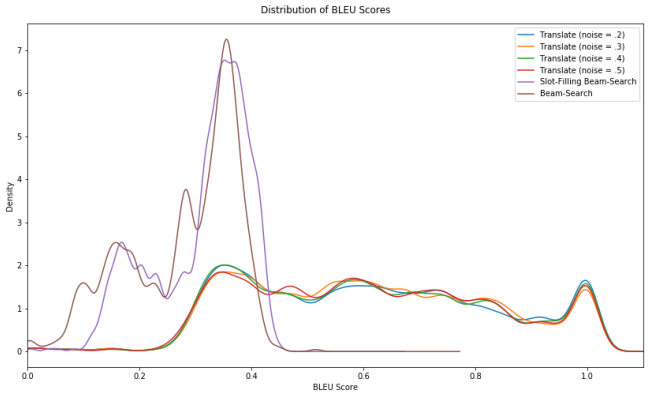

Figure A.34.: Verne: Distribution of BLEU Scores

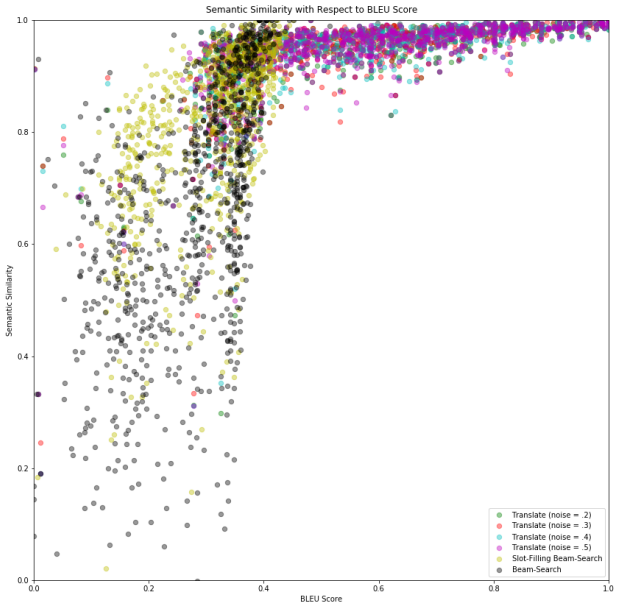

Figure A.35.: Verne: Semantic Similarity with Respect to BLEU Score

	N	M_{A0}	M_{A1}	M_d	SD_d	t_A	p_A
Translate (noise = .2)	908	0.241	0.329	0.088	0.240	11.069	.000
Translate (noise = .3)	898	0.241	0.333	0.092	0.239	11.530	.000
Translate (noise = .4)	897	0.243	0.335	0.092	0.247	11.111	.000
Translate (noise = .5)	896	0.241	0.336	0.095	0.249	11.417	.000
Slot-Filling Beam-Search	822	0.244	0.196	0.048	0.373	3.724	.000
Beam-Search	897	0.237	0.403	0.166	0.468	10.611	.000

Table A.19.: Verne: Paired Samples t-Test Summary for Differences in Authorship Scores

	N	M_S	SD_S	t_S	p_S
Translate (noise = .2)	908	0.947	0.071	83.163	.000
Translate (noise = .3)	898	0.948	0.068	86.501	.000
Translate (noise = .4)	897	0.947	0.072	82.052	.000
Translate (noise = .5)	896	0.947	0.070	84.529	.000
Slot-Filling Beam-Search	822	0.827	0.140	15.679	.000
Beam-Search	897	0.671	0.229	0.000	1.000

Table A.20.: Verne: t-Test Summary for Semantic Similarities ($H_0 : M < .75$)

	N	M_B	SD_B	t_B	p_B
Translate (noise = .2)	908	0.613	0.224	18.495	.000
Translate (noise = .3)	898	0.611	0.217	19.252	.000
Translate (noise = .4)	897	0.610	0.221	18.883	.000
Translate (noise = .5)	896	0.611	0.219	19.008	.000
Slot-Filling Beam-Search	822	0.313	0.084	148.336	.000
Beam-Search	897	0.279	0.098	144.028	.000

Table A.21.: Verne: t-Test Summary for BLEU-Scores ($H_0 : M > .75$)