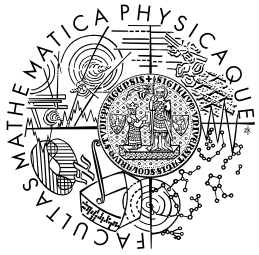


FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Bc. Kristýna Neumannová

German Compounds in Transformer Models

Institute of Formal and Applied Linguistics

Supervisor of the master thesis: doc. RNDr. Ondřej Bojar, Ph.D.

Study programme: Computer Science

Study branch: Language Technologies and Computational Linguistics

Prague 2023

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I would like to express my gratitude to my supervisor, doc. RNDr. Ondřej Bojar, Ph.D. for supervising this thesis and giving me advice. I also would like to thank my family and friends for supporting me during my studies.

Title: German Compounds in Transformer Models

Author: Bc. Kristýna Neumannová

Institute: Institute of Formal and Applied Linguistics

Supervisor: doc. RNDr. Ondřej Bojar, Ph.D., Institute of Formal and Applied Linguistics

Abstract: German is known for its highly productive word formation processes, particularly in the area of compounding and derivation. In this thesis, we focus on German nominal compounds and their representation in machine translation (MT) outputs. Despite their importance in German text, commonly used metrics for MT evaluation, such as BLEU, do not adequately capture the usage of compounds. The aim of this thesis was to investigate the generation of German compounds in Transformer models and to explore the conditions that lead to their production. Our analysis revealed that MT systems tend to produce fewer compounds than humans. However, we found that due to the highly productive nature of German compounds, it is not feasible to identify them based on a fixed list. Therefore, we manually identified novel compounds, and even then, human translations still contained more compounds than MT systems.

We trained our own Transformer model for English-German translation and conducted experiments to examine various factors that influence the production of compounds, including word segmentation and the frequency of compounds in the training data. Additionally, we explored the use of forced decoding and the impact of providing the model with the first words of a sentence during translation. Our findings highlight the importance of further research in developing MT models that are better suited to producing compounds in line with human translation.

Keywords: Transformer, Machine Translation, German compounds, Machine translation quality

Název: Německé složeniny v modelech typu Transformer

Autor: Bc. Kristýna Neumannová

Ústav: Ústav formlání a aplikované lingvistiky

Vedoucí práce: doc. RNDr. Ondřej Bojar, Ph.D., Ústav formlání a aplikované lingvistiky

Abstrakt: Němčina je známá svou velmi produktivní slovotvorbou, zejména v oblasti kompozice a derivace. V této práci se zaměřujeme na německé nominální složeniny a jejich zastoupení ve výstupech strojového překladu. Navzdory jejich důležitosti v německých textech, běžně používané metriky pro hodnocení kvality překladu, jako je BLEU, nedokážou použití složenin dostatečně zachytit. Cílem této práce bylo zkoumat generování německých složenin v modelech typu Transformer a prozkoumat faktory, které vedou k jejich tvorbě. Zjistili jsme, že strojové překladové systémy produkují méně složenin než lidé. Také se ukázalo, že kvůli velmi produktivní povaze německých složenin není možné je identifikovat na základě fixního seznamu. I po ručním vyhledání nových kompozit jich lidské překlady obsahovaly více než strojové.

Natrénovali jsme vlastní model typu Transformer pro překlad z angličtiny do němčiny, abychom to mohli zkoumat různé faktory, které ovlivňují produkci složenin, včetně segmentace slov a frekvence složenin v trénovacích datech. Dále jsme experimentovali s vynuceným dekódováním (forced decoding) a zjišťovali, jak se změní výstup systému po poskytnutí prvních slov překládané věty. Naše výsledky zdůrazňují důležitost dalšího výzkumu v oblasti strojového překladu, aby se byly překladové systémy schopny lépe přiblížit lidskému překladu a generovat více složenin.

Klíčová slova: Transformátor, Strojový překlad, Německá kompozita, Kvalita strojového překladu

Contents

Introduction **3**

1 Background & Related Work **5**
1.1 Compounds in SMT . . . 5
1.1.1 Compound Splitting . . . 5
1.1.2 Generation of Compounds . . . 6
1.2 Neural Machine Translation . . . 8
1.2.1 Transformer Architecture . . . 8
1.2.2 Rich Morphology in NMT . . . 9
1.2.3 Compounds in NMT . . . 10

2 Data and Tools **11**
2.1 Data . . . 11
2.1.1 GermaNet . . . 11
2.1.2 WMT21 . . . 11
2.2 Tools . . . 14
2.2.1 FAIRSEQ . . . 14
2.2.2 Lemmatization . . . 15
2.2.3 UDPipe . . . 15
2.2.4 Other Tools . . . 16

3 German Compounds in English-German Translations **17**
3.1 Compounds in WMT21 Translations . . . 18
3.2 Compounds in WMT21 Training Data . . . 20
3.3 Newly Created Compounds . . . 22
3.3.1 Words not Attested in the Training Data . . . 23

4 Training Own Transformer Model **27**
4.1 Preprocessing of Data . . . 27
4.2 Training Setup . . . 27
4.3 Overall Translation Quality . . . 28

5 Compounds in Our Transformer Model **30**
5.1 Analysis of Compounds Appearance . . . 30
5.2 Aspects Affecting Compounds Generation . . . 32
5.2.1 Novel Words . . . 33
5.2.2 Compound Production vs. Frequency . . . 34
5.3 Compounds as Inference Shortcuts . . . 35
5.4 Forced Decoding Using Prefixes . . . 36
5.4.1 Hinting towards More Compounds? . . . 36
5.4.2 Just One Word Hint Sufficient? . . . 38
5.4.3 Discussion . . . 41

6 Conclusion **43**

Bibliography **46**

List of Figures **50**

List of Tables **51**

List of Abbreviations **52**

A Attachments **53**
A.1 Comparison of Lemmatization Methods 53
A.2 Beam Search Example (Original) 54
A.3 Beam Search Example (Forced Decoding) 55

Introduction

Estimating the quality of machine translation is a challenging task regularly tackled by researchers. Various evaluation methods have been developed for this purpose. The most widely used automatic metric is still the BLEU score [Papineni et al., 2002] which measures the n-gram similarity of the translation output to one or more human reference translations. Another automatic metric for measuring translation quality is, for instance, ChrF [Popović, 2015]. The ChrF metric uses character n-grams instead of word n-grams. However, string-based metrics are not ideal since they measure only the surface similarity to the reference and fail to capture the meaning of the sentence or other subtle phenomena that can also influence the perceived quality of the translation. One such phenomenon is the presence or absence of compound words, a particular grammatical construction that is frequent in German. This thesis focuses on compound words, and how they occur in human and machine translations.

German has a highly productive word formation system mainly through compounding and derivation, especially for nouns [Barz, 2016, p. 2388]. In this thesis, we study German nominal compounds, which mostly consist of two constituents that are either complex or simple stems. The compounds in German are right-headed which means that the second element determines the morphosyntactic properties of the formed word. Additionally, semantically empty elements, called linking elements, can be added to the first stem of the compound. [Barz, 2016, p. 2390]

Using compounds instead of multi-word expressions is a soft phenomenon related to text style, which can affect the perceived quality of the text. Native speakers can even form new compound words to fulfil the needs requested by a particular dialogue or discourse situation. We believe that machine translation systems, operating on subword units, are able to produce complex words like humans, even if they were not included in the training data. In this thesis, we examine Transformer-based models in terms of how often and in which situations they produce compounds in translations from English to German and if and how we can influence that.

We know that splitting and determining German compounds is a complex task. Therefore, we relied on a list of compounds extracted from the German adaptation of WordNet called GermaNet [Henrich and Hinrichs, 2011]. Operating on a closed list of compounds may provide an advantage for the analysis. Considering that the use of compounds is a stylistic matter, the exact list provides us with the possibility to group the observations of the phenomenon.

In the thesis, we study several aspects of the data and models concerning the production of German nominal compounds. One aspect of estimating the quality of translations is the number of appearances of compounds in translations, and we set this metric as our main measure.

Firstly, we compared the counts of compounds in human translations with state-of-the-art machine translation system outputs for the English-German language pair. We examined their correlation with the general translation quality measured by the BLEU score. Then, we trained our own Transformer model to explore its behaviour regarding compound production on a fixed training set.

Since the weights and inner states of the model are hard to interpret, clustering examples (or sentences) that contained compounds helped us explore the similarities and differences among different model versions. We analyzed the factors that influenced compound production. Furthermore, we also searched for words in the output that are not contained in the training data, compared the counts of produced compounds within the training epochs of the system, and performed further analysis concerning the compound counts in the outputs. Several training setups that differed in preprocessing of the corpora were compared. The source code and the outputs of the performed analysis are available on GitLab.[1] The repository contains scripts and outputs of the compound analysis, training, and preprocessing scripts and additional code for experiments on our Transformer.

The structure of the text is as follows: Chapter 2 describes the data and tools used for compound analysis and Transformer training. Chapter 3 provides a study of the appearance of German compounds in state-of-the-art English-German translations. Chapter 4 specifies the parameters and data used for the training of our own Transformer model. The subsequent analysis of the outputs of our model is described in Chapter 5. We investigate the production of compounds and their adaptations and focus on aspects that can influence the production.

[1] https://gitlab.mff.cuni.cz/neumankr/master-thesis

1. Background & Related Work

German compounds are complex words that occur in every German text. However, they pose a challenge for natural language processing (NLP) applications such as morphological analyzers or machine translation (MT) systems. There have been several attempts to find and process compounds to improve the quality of these applications. Most of the previous work on MT dealing with German compounds was done in statistical machine translation (SMT). We found only a few papers, see below, about German compounds in neural machine translation (NMT), almost all of which were published before the introduction of the Transformer model [Vaswani et al., 2017]. The Transformer model outperformed the previous NMT systems and became the leading neural network architecture not only in NLP but also in a broad range of other applications. Our work focuses on the production of German compounds in Transformer models, a topic that has not been adequately studied yet. We describe previous work on German compounds in SMT (Section 1.1) and NMT (Section 1.2).

1.1 Compounds in SMT

The most common approaches to SMT operated on whole words. Therefore, they did not handle morphologically rich or compounding languages very well. This led to the introduction of several methods to improve SMT quality for compounding or inflectional languages. One such method is compound splitting which aims to split complex words into parts that can help translate unknown words. We can translate their parts after splitting them and then join them. Morphological analysis of a text or determining compounds in a text or a set of words can aid in this process. We describe the approaches to compound splitting in Section 1.1.1. These methods dealt with improving MT quality from compounding languages.

There were other approaches that focused on translation into compounding languages and performed post-processing of translations and merging words into compounds. The data passed to MT systems were often pre-processed with a word splitter, and the compound generating part was then employed after the translation using compound merging strategies. It is not possible to fully separate approaches dealing with compound splitting and compound generation because they were often combined. However, we present work focusing mostly on merging strategies in Section 1.1.2

1.1.1 Compound Splitting

One of the first empirical methods for splitting compounds was introduced by Koehn and Knight [2003]. In their paper, the authors attempted to split compounds into parts that had been separately observed in the training data. They focused on improving MT quality and used a metric that considered the frequency of the constituents in the training data. Another factor that influenced the splitting of compounds was the number of similar words (with the same component) that were split. For example, the German word *Grundrechte* was better to be split because the first constituent *Grund-* was a frequent part of compounds. Er-

rors in split prefixes and suffixes were resolved using limitations on the part of speech (POS).

Schmid et al. [2004] presented SMOR, a German morphological analyzer that focused on productive derivation and compounding processes in German. The analyzer was implemented as a finite-state transducer and had an impact on other work in German morphology and compound splitting. For example, Henrich and Hinrichs [2011] used an adapted version of SMOR to improve the German compound splitting algorithm for determining the constituents of compounds in GermaNet. They combined an updated SMOR compound splitter with other splitters, such as a pattern-matching-based splitter that considers all potential modifiers and heads, along with linking elements. This approach generated a list of nominal German compounds that were used for our analysis.

Besides morphology-based data-driven methods for compound splitting, unsupervised or semantic-based approaches also exist. For example, Sugisaki and Tuggener [2018] introduced an unsupervised method for compound splitting based on the productivity of morphemes. This approach distinguishes between bound and free morphemes. Free morphemes have the ability to stand alone as words, while bound morphemes appear only as parts of words in the text. They computed a probability of morpheme boundedness, which is the ratio between the counts of bound and free morphemes. The lexicon used for this approach was extracted from a giga web corpus.

Another approach based on semantic representations was presented by Daiber et al. [2015]. They visualized compound words and their constituents in vector space for better intuition of how are their representations distributed in relation to each other. Compounds with the same head, which is the second part of the compound carrying morphosyntactic properties, tended to be close in the vector space, which did not necessarily hold for compounds with the same modifier, which is the first part of the compound. The authors proposed a method for extracting compounds based on this observation. They recursively extracted all possible modifiers (all meaningful prefixes that also left meaningful suffixes) and then created prototypical vectors for them. For each extracted modifier-head pair, the directional vector was computed by subtracting the head vector from a compound vector. These retrieved modifier vectors were then added to all vectors in the dictionary of heads. If the resulting vector was similar to some word, it was then considered a compound. The authors evaluated their splitter based on the described prototypes on the GermaNet list of compounds. They achieved 27.4% accuracy and 58.4% coverage on the test set. This compound splitting method has been shown to improve SMT systems with the BLEU score on the WMT15 News test set improving by 0.6 BLEU compared to the baseline translation from the Moses decoder Koehn et al. [2007].

1.1.2 Generation of Compounds

Although we could not entirely separate methods dealing with splitting and merging compounds, in this section, we present approaches that explored the generation of compounds or merging strategies for them in more detail. Popović et al. [2006] focused on both German-English and English-German translation. For translation into German, they collected all German words consisting of two or

more components and decided whether to split them based on the frequency of the whole expression and its parts. They proposed three different methods for English-German translation, one of which dealt with splitting and merging German compounds. They divided the compounds into parts as described in Section 1.1.1 for German-English translation. After that, the authors trained the MT system and let it translate the input text. The resulting text was post-processed, i.e., the compounds were merged. The merging method was based on corpus statistics. They extracted a list of German compounds and their constituents from the training corpus. The merging algorithm then searched for words that followed each other in the output text in this list. The sequences of words that were found as constituents of compounds in the list were then merged into a compound. This method enabled the production of only compounds that are composed of known compound parts in the training corpus.

Stymne [2009] investigated merging strategies for compounds on the target side of SMT including the method of Popović et al. [2006]. She focused on translating into German. As in the previously cited article, compounds were first split in the training data, and after translation, the parts were merged into complete compounds. Stymne compared several merging algorithms in her paper. The first algorithm merged tokens seen in the training data as compound parts (the method called "word-list" based on Popović et al. [2006]). The second dealt with merging tokens marked with a special symbol (the method called "symbol"). The third merged all tokens with a compound POS tag ("POS-match"). These three main groups of algorithms also had some modifications. These methods were evaluated in two ways: the overall quality of the translation was evaluated, and the performance of merging algorithms was analysed (according to the number, type and quality of merges). It was shown that merging strategies could improve SMT quality; however, none of the investigated algorithms reached the number of compounds in the human-translated reference.

Stymne and Cancedda [2011] built on Stymne's previous work and proposed a method for compound merging that viewed the task as sequence labelling. The words were labelled as to whether they should be joined or not. Further improvements were made by combining and enhancing the heuristics for merging described by Stymne [2009].

As composition is a highly productive word formation process in German, MT systems should be able to produce unseen compounds to deal with it. Cap et al. [2014] focused on improving SMT through the compound synthesis in English-German translation. Their systems operated on English-German translation. To produce unseen compounds, the authors preprocessed the training data by splitting words into parts and computing the frequency statistics of these parts. The output of the MT system was then post-processed using these statistics. Potential compounds were merged based on frequencies of the bigram, the second part as the head of a compound, and the first part as the modifier of a compound from the training data. If the target language features were insufficient, source language features such as alignment or POS features were used. As the automatic evaluation using the BLEU metric did not show significant improvements, they provided a review of the compounds by identifying them manually, aligning them with English source text, projecting these English counterparts, and then annotating the resulting tuples of German compounds and their English counterparts.

Their method generated 100 more compounds (750 in total) than the baseline Moses decoder Koehn et al. [2007]. Many of the found compounds were correct translations of the source text even if they were not all affirmed by the reference translation.

1.2 Neural Machine Translation

Neural machine translation (NMT) replaced the previous approaches between the years 2014–2017. In this brief summary, we skip the previous models based on recurrent networks and focus only on the Transformer architecture (1.2.1), subword units (1.2.2), and compounds in NMT (1.2.3).

1.2.1 Transformer Architecture

Vaswani et al. [2017] introduced a Transformer architecture which surpassed the performance of the previous state-of-the-art recurrent and convolutional networks in MT. Like other sequence-to-sequence models, the Transformer has an encoder and decoder. The encoder produces a representation of an input sequence, which is then fed to the decoder to iteratively generate an output sequence. The output is then transformed into the final output, which in the case of translation means producing words in the target language based on vocabulary indices.

The key idea behind the Transformer architecture was the self-attention mechanism, which enabled interactions between all positions in the input or output, even if they were very distant. This mechanism was implemented by the scaled dot product of queries with keys obtained from the input, which were (after applying the softmax function) multiplied with values that were also extracted from the input. The authors proposed to use multiple parallel self-attention computations and constructed a multi-head attention mechanism. This mechanism was utilized in both the encoder and the decoder of the model. Each encoder layer consisted of a self-attention sub-layer and a fully-connected feed-forward sub-layer, with normalization and residual connections employed after each sub-layer. In addition, the decoder included a third sub-layer that provided multi-head attention over the output of the encoder. The decoder self-attention sub-layer was modified to ensure that the predictions for a particular position only depend on previous predictions. The decoder of the sequence-to-sequence model generates a final sequence of tokens, and in each step of generation, one token is added to the previous sequence. Since the number of possible words to add to the translated sentence in each step of decoding is very large, the path through the entire search space must be reduced to save the memory and speed requirements of the algorithm. Sutskever et al. [2014] introduced a general approach to sequence-to-sequence learning for neural networks. The decoder search was equipped with a left-to-right beam search that kept a small defined number B of partial hypotheses. B is called beam size or beam width. The partial hypothesis is considered to be a prefix of some potential translation. In each step of decoding, they extended each partial hypothesis in the beam with every possible word in the vocabulary. After that, the partial hypotheses with lower probabilities were discarded, so only the B best hypotheses remained in the beam. The Transformer model utilized this search algorithm for decoding.

Holtzman et al. [2019] investigated several decoding strategies, including beam search. They discovered that maximization-based decoding algorithms, such as beam search, lead to degeneration, meaning that the output text is dull, incoherent, or repetitive. However, their work focused mainly on open-ended generation, which includes tasks such as conditional story generation and text continuation. Since translation is a directed-generation task and its output is tightly scoped by the input, repetitions are not so problematic. The authors examined several aspects of text generation that were influenced by the chosen decoding algorithm. One of them was vocabulary usage and text perplexity. They discovered that the text generated by maximization is too probable, which means that it lacks diversity and divergence in vocabulary usage. This differentiates machine-generated text from human-written text. Their statistics revealed that high values of beam size led to vocabulary usage more similar to human distribution; however, these texts often had a high variance in likelihood and were, therefore, less coherent. The perplexity of the text generated by maximization-based algorithms was found to be much lower than it was for human text. To overcome these issues, the authors proposed a new decoding strategy called Nucleus Sampling, which utilized sampling with a dynamically changing parameter.

The search space of all possible sentences is huge, so the algorithm uses pruning which can lead to various errors in decoding. We distinguish three types of errors: search error, modelling error, and compound error. When the model scores better on an incorrect (or undesirable) sentence than the correct one, we call it a modelling error – the model did not fulfil our wishes. This allows the model to make a search error where a better-scored sentence exists but was not found during the search. The compound error is the combination of these two errors.

1.2.2 Rich Morphology in NMT

Morphological analysis, compound segmentation, and merging are easily implemented in SMT. Nevertheless, NMT has surpassed SMT in performance and is still evolving. Several attempts have been made to address the complex morphology of inflectional or compounding languages in NMT. Although it is challenging due to the non-intuitive nature of neural networks' hidden parameters, dealing with rich morphology is essential to improve translation quality for languages such as German or Czech.

Translating into inflected and compounding languages requires the ability to generate words from an extensive vocabulary. However, dealing with a large vocabulary in NMT is very computationally demanding, so the dictionary size needs to be limited. However, translation is an open-vocabulary problem and the MT system should be able to produce rare and unknown words. Sennrich et al. [2016] introduced a simple and effective approach based on the idea that various categories of words, such as compounds and other morphologically complex words, names, or loanwords, can be translated using smaller units than words. These units are called subwords. The authors proposed a word segmentation method based on the Byte Pair Encoding (BPE) compression algorithm [Gage, 1994]. The original algorithm iteratively replaced the most frequent bytes in a sequence with another unused byte. The proposed segmentation method merges characters or

character sequences instead of pairs of bytes. The initial vocabulary included all characters plus the special end-of-word symbol. Then, the algorithm iteratively replaces each occurrence of the most frequent symbol pair with a new symbol for the pair. Each merge produces a new symbol representing a character n-gram (or possibly a whole word). The algorithm iterates until the desired size of the vocabulary is reached. The BPE word segmentation method has led to big improvements in the translation of rare and unseen words, and it has become a prominent method for segmentation in NMT.

Tamchyna et al. [2017] proposed a two-step translation system to incorporate the morphological features of target side words. In the first step, they used an encoder-decoder NMT system with BPE to produce a sequence of interleaving lemmas and morphological tags. In the second step, they applied a morphological generator to construct the final inflected words. Although their work mainly focused on English-Czech translation, they also carried out experiments for the English-German pair. For German, they used a morphological analyzer instead of a simple lemma and morphological tag tuple as in Czech to cover productive word formation processes such as compounding.

Weller-Di Marco and Fraser [2020] extended the lemma-tag generation approach from Tamchyna et al. [2017] and implemented source-side word segmentation based on statistics retrieved from tagged and lemmatized data. Their method operates on suffixes and prefixes and was investigated using English-German translation implemented by the Transformer model. The evaluation was again performed using the BLEU metric.

1.2.3 Compounds in NMT

Weller-Di Marco and Fraser [2020] integrated a compound splitter for German into their previously described translation system. The splitter was based on morphological analysis and relied on a morphological analyzer from Koehn and Knight [2003]. Their compound splitter was similar to Weller-Di Marco [2017], which combined a basic frequency-based approach with a form-to-lemma mapping. They assumed that components of compounds could possibly be inflected, and lemmatization of them would solve it. However, adding a compound to their two-step translation did not show significant improvements.

As described in the previous section, the used word segmentation technique influences the ability of the model to deal with complex words and the rich morphology of inflectional and compounding languages. Huck et al. [2017] investigated word segmentation strategies that incorporate more linguistic knowledge than the widely used BPE. One of the described strategies involved compound splitting and provided top-down segmentation that considers the frequency of the components, in contrast to BPE, which operates bottom-up. Compound splitting combined with suffix splitting improved BPE word segmentation in English-German translation, as evaluated by the BLEU score.

2. Data and Tools

2.1 Data

In this section, we present the data that was used to analyse the presence or absence of German compounds in English-German translations, as well as the fixed dataset that was used to train our Transformer model. The compounds included in the systems' outputs and in the training data were determined based on a published list of compounds extracted from GermaNet. For our model, we used a training and testing dataset from the Sixth Conference on Machine Translation (WMT21).[1] The outputs from submitted systems to the conference were explored in terms of translation quality and the number of compounds contained within them.

2.1.1 GermaNet

GermaNet is a German word net that preserves the database format and structure of Princeton WordNet 1.5. However, GermaNet does not provide a translation of WordNet. It is an independently created word net with its own concepts and features that focus on the German language. GermaNet was composed of lexicographic resources, such as Wehrle and Eggers (1989), and was manually assembled based on corpus frequencies [Kunze and Lemnitzer, 2007]. Additional resources have been added over time.

Nouns, adjectives, and verbs are the most important categories of words in the word net. The central representation concept in GermaNet is the synset that groups synonyms of a given topic, such as *Streichholz* and *Zündholz* (matches). The word net captures semantic relations between the synsets and also between synonyms in one synset [Kunze and Lemnitzer, 2007]. The authors distinguished two types of relations: lexical, such as synonymy and antonymy, and conceptual, like hyponymy, hypernymy and others.

As mentioned above, we used a fixed list of nominal German compounds extracted from GermaNet [Henrich and Hinrichs, 2011] (version v17.0 last updated in June 2022) to determine German compounds. The list contains 115,563 nominal German compounds with information on how they are split into two parts: the head and the modifier of the compounds. The first part modifies the meaning of the second part (the head), which carries the morphosyntactic features of the entire word [Barz, 2016, p. 2390]. Compounds with more than two constituents can be recursively split by finding the split of its components in the GermaNet list.

2.1.2 WMT21

We used a dataset provided by the Sixth Conference on Machine Translation (WMT21) [Akhbardeh et al., 2021] and tested our hypotheses on the outputs of systems submitted to the conference. Our own Transformer model was trained

[1] https://www.statmt.org/wmt21/

using a closed set of parallel training data and then tested on the Newstest2021 test set provided by the authors of the metrics task at the conference.

Training Data

We trained our Transformer model for English-German translation using seven parallel corpora which were the same as those used for constrained systems submitted to WMT21. The constrained systems did not use any additional data except for the given corpora for training. In the provided data, each sentence was provided on a separate line, and we split it into tokens using the Moses tokenizer (see Section 2.2.4). The number of sentences and English and German tokens in the datasets is shown in Table 2.1. We set aside 10% of the data for validation, as suggested by the translation example from FAIRSEQ (see Section 2.2.1). Therefore, only 90% of the data was used for training.

corpus	**sents**	**EN tokens**	**DE tokens**
ParaCrawl v7.1	82,638,202	1,644,732,036	1,588,959,138
WikiMatrix	5,473,201	107,836,063	103,902,159
Common Crawl corpus	2,399,123	59,894,034	55,956,928
Europarl v10	1,828,521	50,964,634	48,517,316
Tilde Rapid corpus	1,631,639	1,631,639	26,915,003
Wiki Titles v3	1,474,203	3,966,045	3,563,107
News Commentary v16	398,981	9,979,602	10,059,964

Table 2.1: WMT21 Training corpora - number of sentences and tokens

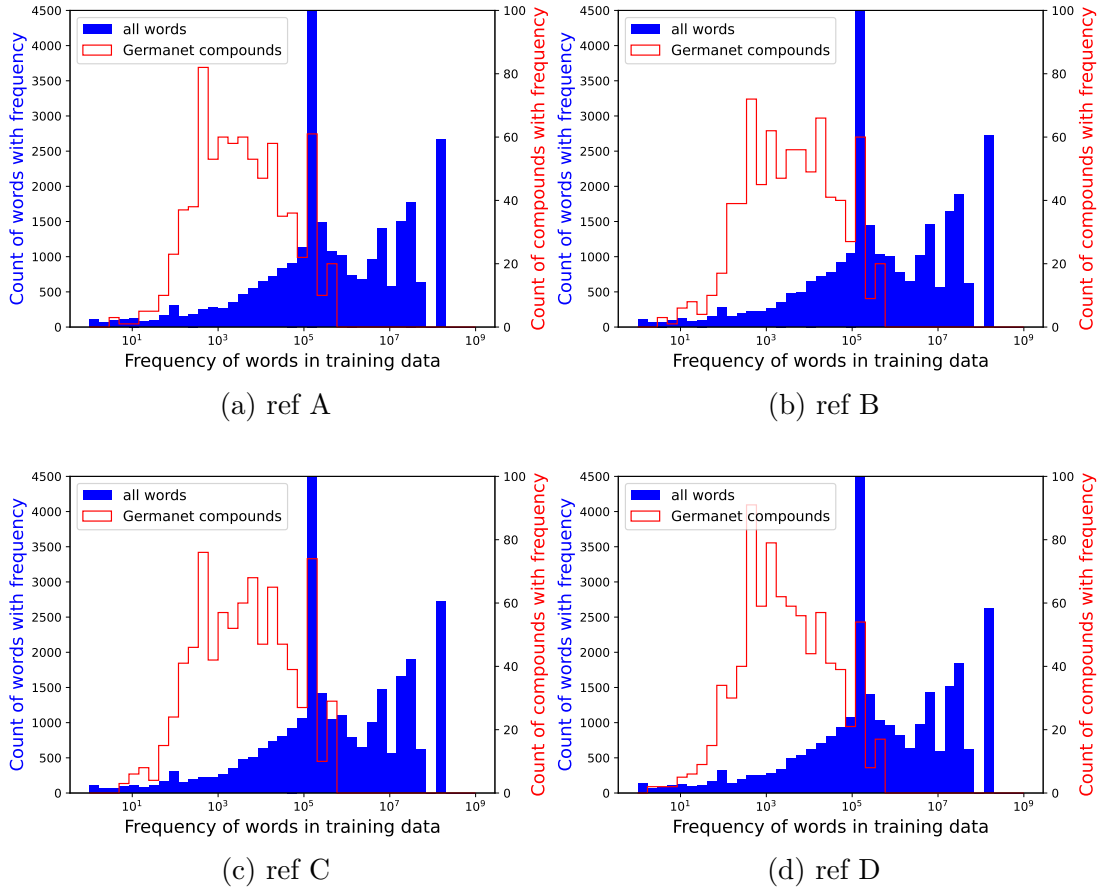
Test Data

To compare human translations to outputs of state-of-the-art systems and evaluate our model, we used the WMT21 test set [Akhbardeh et al., 2021]. The source texts for this dataset were retrieved from online news sites. The test set comprises around 1,000 sentences for all languages (1,002 for en-de). The authors of the test set guaranteed that the sentences were originally from the source language and then translated into the target language. The sources of the English articles used in the test set are listed in Figure 2.1.

```
ABC News (5), Al Jazeera (1), All Africa (2), BBC (4), Brisbane Times (3), CBS LA (1), CBS
News (3), CNBC (1), CNN (1), Daily Express (4), Daily Mail (1), Egypt Independent (3), Fox News (2),
Guardian (6), LA Times (1), London Evening Standard (2), Metro (1), NDTV (7), New York Times (2),
RTE (1), Russia Today (5), Seattle Times (4), Sky (1), The Independent (1), The Sun (2), UPI (1),
VOA (1), news.com.au (1), novinite.com (1)
```

Figure 2.1: Composition of English test-set (number of articles)

Professional translation agencies performed the reference translations. Considering that English-German is a highly attractive language pair, it received special attention. A different translation agency provided a second reference, labelled "B"; however, it was found to be a post-edited version of one of the submitted systems, so it was discarded from the conference. Since we do not rely on the BLEU score in most of our experiments, we left the second reference in the



(a) ref A (b) ref B

(c) ref C (d) ref D

Figure 2.2: Histogram of lemmas present in references according to frequencies in training data

test set for compound analysis. However, we did not use it for BLEU scoring. The third reference translation was sponsored by Microsoft, labelled "C". The metric task organizers then provided a fourth reference, labelled "D". [Akhbardeh et al., 2021]

The plot in Figure 2.2 shows the comparison of training and test data distributions for each reference separately. The x-axis shows the frequencies of words in the training data, and the y-axis shows the count of tokens in the reference belonging to a particular frequency interval. The blue bars represent all words, while the red bars represent specifically compounds (secondary y-axis). The statistics were computed on lemmatized texts.

The distributions of all words are very similar for all four references, but the counts of compounds differ. The reference translations "C" and "D" contain the most compounds, while the reference "B" has the least. It may be due to the fact that the reference "B" is a postedited version of an MT system. It is interesting to note that the most frequent lemmas, which belong to the last bin, are *"–"* and *"der"* (definite article in German).

```
C-BUPT_rush          UC-metricsystem1
C-eTranslation       UC-metricsystem2
C-HuaweiTSC          UC-metricsystem3
C-ICL                UC-metricsystem4
C-Manifold           UC-metricsystem5
C-Nemo               UC-Online-A
C-nuclear_trans      UC-Online-B
C-P3AI               UC-Online-G
C-UEdin              UC-Online-W
C-UF                 UC-Online-Y
C-WeChat-AI          UC-VolcTrans-AT
UC-Facebook-AI       UC-VolcTrans-GLAT
UC-happypoet
```

Figure 2.3: Systems participating on WMT21 for English-German translation (C marks constrained systems and UC unconstrained)

System Outputs

We downloaded the outputs of the systems submitted to WMT21 for the English-German language pair to compare their state-of-the-art results to human translation and our own results. We focused on compound counts, which we chose as the main criterion for our comparison. However, not all provided systems used only the given fixed training set, so we distinguished constrained (C) and unconstrained (UC)systems and marked them in our statistics. All systems are listed in Figure 2.3.

2.2 Tools

This section describes the tools and frameworks used for our analysis and Transformer training. WE wrote the majority of the code in Python 3 or Bash. We selected FAIRSEQ [Ott et al., 2019] as the framework for training and evaluating Transformers, as explained in Section 2.2.1. Prior to identifying compounds in the outputs, we had to lemmatize the text. In Section 2.2.2, we explain several different lemmatization methods, that we used. Additionally, we used some minor tools during our analysis, which are described in Section 2.2.4.

2.2.1 FAIRSEQ

The FAIRSEQ toolkit [Ott et al., 2019] is an open-source tool used for sequence modelling and allows researchers to train and evaluate their custom models for text-generating tasks such as translation, language modelling and summarization. It is written in PyTorch and designed to run on multiple GPUs.

The authors also provide command-line tools to launch Python training and evaluating scripts. The toolkit repository contains several sample pipelines that implement specific research papers. For our preprocessing, we used a script that was inspired by the provided WMT14 English to German example.[2]

We utilized the *fairseq-preprocess* script to binarize the train, test, and validation data. Our Transformer model was trained using the *fairseq-train* script and we generated the outputs of the models using the *fairseq-generate* script.

[2]https://github.com/facebookresearch/fairseq/tree/main/examples/translation

We adapted the generation script to output not only the final hypotheses (complete translations) but also candidate hypotheses for each generation step. The parameters used for training are further described in Chapter 4.

2.2.2 Lemmatization

To obtain accurate results, we had to normalize the translations (human and MT) to identify German compounds in the translated text. We initiated the process by lemmatizing the German text using the Spacy framework.[3] We downloaded the small model for German, *de_core_news_sm* for Spacy, and lemmatized all human and system translations of the Newstest2021 test-set. Additionally, we lemmatized the German part of the training data to detect new words in the systems' outputs and generated statistics about their frequencies in the training data.

However, we encountered some errors in lemmatization, particularly for long German compounds – many plural nouns were not normalized and not all genitive nouns with ending *-s* or *-es* were recognized. To address these issues, we developed a simple rule-based normalization method that primarily focused on genitive endings. This method employed external tools for tokenization, as described in Section 2.2.4, and was advantageous as it correctly processed numerous complex nouns. Since our analysis centered on German nouns, we did not concentrate on normalizing other words in this method. We found this simple rule-based method to be an excellent complement to Spacy lemmatization since both techniques identified supplementary sets of German compounds.

However, after evaluating both methods, we decided not to combine them and chose the UDPipe 2 [Straka, 2018] lemmatization method instead because it performed better. The description of this method is provided in the subsequent section.

2.2.3 UDPipe

UDPipe 2 is a Python toolkit for tagging, lemmatization, and syntactic analysis developed by Straka [2018]. UDPipe operates on CoNLL-U[4] inputs. The second version of UDPipe is implemented as a client-server-based application. It can be used either via REST service or by compiling a server locally and then sending requests to it. Since we wanted to lemmatize the German texts of the entire training data, we compiled the server locally. The server and client scripts are available on GitHub.[5] We ran UDPipe and Wembedding [6] servers on separate GPUs to achieve better performance in lemmatization of complex words. The Wembedding server provides the computation of contextualized embeddings.

In a small manual examination, we found that the pre-trained German GSD model[7] from the 2.10 version of Universal Dependencies models[8] is the best option for lemmatization of complex compounds. The GSD model performed better

[3]https://spacy.io/
[4]https://universaldependencies.org/format.html
[5]https://github.com/ufal/udpipe/tree/udpipe-2
[6]https://github.com/ufal/wembedding_service
[7]https://universaldependencies.org/treebanks/de_gsd/index.html
[8]https://ufal.mff.cuni.cz/udpipe/2/models#universal_dependencies_210_models

than the other German models for that version, particularly in processing long compounds and it did not omit modifiers from the compounds. It also successfully processed many nouns with genitive and plural endings.

During manual analysis of lemmatization errors based on a comparison of UDPipe with Spacy, we uncovered several inconsistencies between produced lemmas and the normalized compounds in the GermaNet list. The comparison of lemmatization methods in counts found in translations is displayed in Appendix A.1. One of the differences in UDPipe lemmatization was a substitution of the German letter sharp s (*ß*) for *ss* in nouns. Based on these observations, we decided to lemmatize also the list of GermaNet compounds for consistency.

2.2.4 Other Tools

Some outputs were only tokenized for the analysis. We utilized tokenization for our rule-based lemmatization and for further analysis of our Transformer model, for instance for stripping the first n words of a sentence before computing the BLEU score which has to be done on the original output and not on the lemmatized version of it. Tokenization for text normalization was implemented using the NLTK framework with the function *word_tokenize()*.[9]

The FAIRSEQ tool processed its inputs for generation and evaluation with the Moses tokenizer and detokenizer [Koehn et al., 2007]. Therefore, we used the same tokenization method to analyse specific outputs of the tool, specifically for prefix statistics. Consistent tokenization was needed to strip a prefix of a particular length from a sentence. We used the Python implementation of the Moses tokenizer and detokenizer (Sacremoses[10]).

Because the training of an NMT model is limited to a fixed vocabulary size, the vocabulary of the training data had to be reduced. This was achieved by dividing less frequent words into subwords using the BPE method. We used the subwordNMT [Sennrich et al., 2016] implementation to learn and apply BPE.[11]

To assess the general translation quality, we computed BLEU scores for the WMT21 systems' outputs as well as the outputs of our Transformer models. In the thesis, we provide BLEU scores against the first reference and also against all references (without the discarded one). We used the SacreBLEU [Post, 2018] implementation[12] of the BLEU metric.

[9] https://www.nltk.org/api/nltk.tokenize.html
[10] https://github.com/alvations/sacremoses
[11] https://github.com/rsennrich/subword-nmt
[12] https://github.com/mjpost/sacreBLEU

3. German Compounds in English-German Translations

The first part of the thesis analyses the presence or absence of German compounds in English-German translations. As previously stated, identifying German compounds is challenging, so we relied on the list of German compounds from GermaNet (see Section 2.1.1) during the analysis. The study focuses on determining if MT systems produce compounds at similar rates and in the same sentences as human translators. This research may lead to a better understanding of the phenomenon of creating German compounds and improving MT quality in terms of productive composition for Transformer models, which are the subject of the thesis.

We visualised notable sets of compounds to compare two different system outputs and their intersections. Figure 3.1 displays these sets with their numbered intersections. We have two different translations of the test sentences and the compounds comprised in them (blue and red), the compounds from the GermaNet list (green), and compounds that occurred in the training data (violet). The entire rectangle signifies their superset – all possible German compounds. We could theoretically add a set of compounds relevant to the test text, but there is no easy way to add a fifth subset, and besides, this set is not easy to describe or collect.

The only set we know entirely is the set of GermaNet compounds. We can estimate all subsets of it (subsets II, III, IV, V, VI, VII in Figure 3.1). Approximately 3.5%–5% of the GermaNet compounds are not included in the training data (the percentage is biased by the used lemmatization method). We assume that the sets VI, IV, and II will be very small or empty in the constrained systems (systems that were trained using only the fixed set of training data).

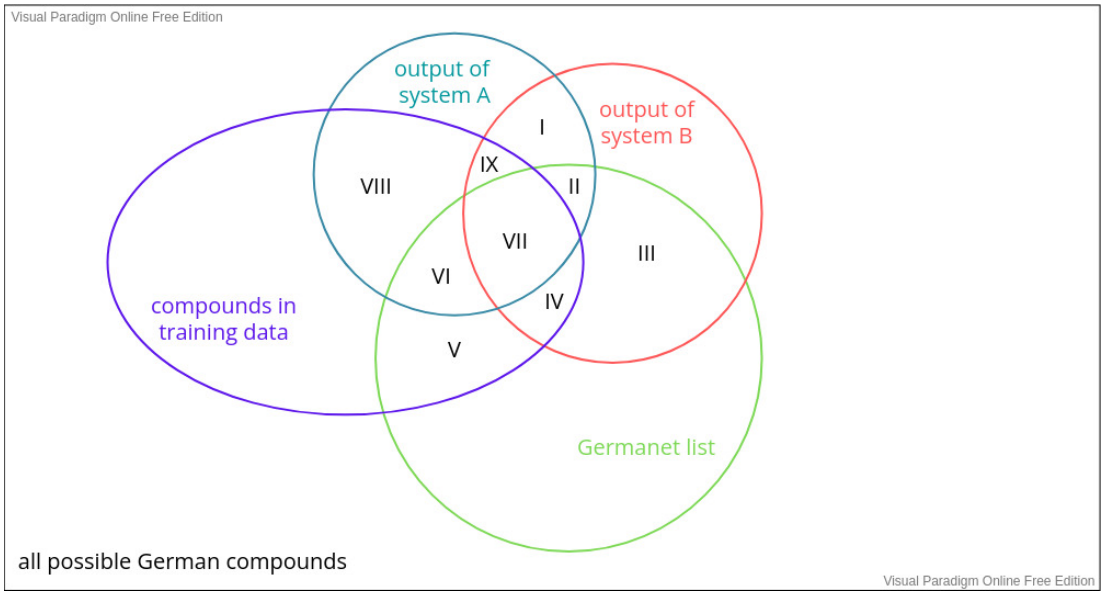


Figure 3.1: Sets of German compounds appearing in outputs of two systems and a list of compounds from GermaNet

Unfortunately, determining the counts of German compounds in the subsets depends on lemmatization. The texts need to be lemmatized before searching for the compounds in them because we only have a closed list of German compounds in a standardized form. The quality of lemmatization affects the counts of the found compounds. In Section 2.2.2, we present the lemmatization methods we experimented with.

In the following sections, we examine the sets described in Figure 3.1. We compare the counts of compounds in different translations from WMT21 with each other and with the training set.

3.1 Compounds in WMT21 Translations

In this section, we focus only on compounds that are contained in the GermaNet list and search for them in WMT21 translations. We compare the counts of compounds found in reference translations and state-of-the-art system outputs, i.e. the sets II, III, IV, VI, and VII from the diagram in Figure 3.1 are investigated. The figure displays the relation of only two systems, but we have more translations to compare. We did not compute the sizes of all subsets for each pair.

We present counts of compounds and counts of sentences that contain at least one compound for each reference and output translation separately. For the constrained systems, we also report the number of compounds and sentences with compounds that also appeared in one or more reference translations in a table (considering only the same sentence it appeared in and not counting multiple occurrences of the same compound in a sentence). The results are sorted by the number of found compounds decreasingly and listed in Table 3.1.

Table 3.1 shows that human reference translations contain more compounds than any other MT systems' outputs. The best reference regarding the compound number is the reference "C", with 955 compounds found in 593 sentences. That is over 100 compounds more than in the best MT system. The source text for all of the translations comprised 1,002 sentences, so more than half of them led to the generation of some compound in the best reference translation. Considering all sentences where at least one human translator used a compound, we get 756 sentences with 995 different compounds. For all translations, we have 898 out of 1,002 sentences where at least one compound occurred.

Considering only the number of produced compounds, the best MT system is the constrained system *Nemo*, with 842 compound occurrences in 559 sentences (see Table 3.1). 87% of the compounds are approved by references. Unconstrained systems that employ extra training data are presumed to have better results than constrained systems. However, two constrained systems, *Nemo* and *UF*, each produced more compounds than any of the unconstrained systems. The worst system, *ICL*, contained 138 fewer compounds than the best MT system and 251 fewer compounds than the best human translation (see Table 3.1).

Although we sorted the table according to the compound count, the sentence count is also important, especially for comparison with reference translations. The same concept can be translated as a different compound, and as such it would not be validated by the reference despite being also correct. Counting all sentences with a compound helps to mitigate this effect. We take this into account by considering four different human translations instead of only one.

system	# compounds	in refs	# sents	in refs
ref-C	955		593	
ref-D	946		591	
ref-A	901		566	
ref-B	878		569	
C-Nemo	842	735	559	511
C-UF	802	710	532	487
UC-metricsystem2	801		533	
UC-Online-B	798		532	
UC-Facebook-AI	796		533	
C-eTranslation	794	696	530	486
UC-VolcTrans-GLAT	792		533	
UC-Online-W	791		533	
UC-metricsystem1	790		530	
UC-metricsystem3	787		518	
UC-metricsystem5	783		531	
C-WeChat-AI	783	707	527	493
UC-VolcTrans-AT	782		531	
UC-Online-Y	776		522	
UC-happypoet	770		526	
UC-metricsystem4	769		515	
C-Manifold	768	666	514	460
UC-Online-A	767		520	
C-nuclear_trans	762	656	514	466
C-HuaweiTSC	761	673	516	473
C-UEdin	758	666	513	466
UC-Online-G	754		516	
C-P3AI	740	655	505	467
C-BUPT_rush	731	627	495	443
C-ICL	704	595	485	426

Table 3.1: Compounds appearance in English-German translations in WMT 21 (counts of all appearances of compounds and counts of sentences with compounds plus its subsets approved by reference translations)

We have calculated BLEU scores for all the systems to compare their overall translation quality with our metric of produced compound counts. The BLEU scores against reference A and against all three references (A, C, D) for all systems are displayed in Table 3.2. While the order of systems based on the BLEU score against reference A does not entirely correspond to the order based on our compound metric (in Table 3.1), the first six systems scored high in both metrics, and the two worst systems placed in the same positions in both metrics. The constrained system *UF* contained many compounds but did not receive a high BLEU score. The results have shown that we could not entirely rely on the BLEU score when dealing with compound production but the correlation between these two metrics is large.

system	BLEU refA	BLEU all
UC-VolcTrans-GLAT	31.34	64.33
UC-Facebook-AI	31.26	62.00
C-WeChat-AI	31.32	60.34
C-Nemo	30.01	58.84
UC-Online-W	29.71	62.64
C-eTranslation	29.59	57.40
C-HuaweiTSC	29.77	57.94
C-UEdin	29.90	57.03
UC-Online-A	29.03	57.09
C-Manifold	29.43	56.03
UC-VolcTrans-AT	29.31	56.47
C-UF	28.47	56.88
UC-metricsystem1	28.29	56.43
UC-metricsystem4	28.55	56.45
UC-Online-B	28.40	56.84
C-P3AI	28.32	55.97
UC-metricsystem2	27.94	54.14
UC-happypoet	27.56	53.58
UC-Online-Y	27.94	53.04
UC-Online-G	27.08	52.65
C-nuclear_trans	27.70	51.93
UC-metricsystem5	26.66	52.23
C-BUPT_rush	26.36	50.57
UC-metricsystem3	25.97	51.12
C-ICL	24.54	46.00

Table 3.2: BLEU scores of WMT21 systems – for reference A and all references excluding B, sorted descending by the score for refA

3.2 Compounds in WMT21 Training Data

In the previous section, we measured the sizes of sets of Germanet compounds in the translations' outputs. Now we focus on the relation of the compounds in the outputs to the training data, again subject to the coverage of GermaNet. As shown in Figure 2.2 the compounds contained in the references have frequencies approximately between 10^1 and 10^4 in the training data.

We collected all examples of compounds found in the sentences of the WMT21 systems' outputs and all four references. The instances were distinguished by the compound and the identifier of the sentence the compound appeared in. Therefore, if there were several occurrences of one compound in the sentence, we counted it only once. Still, we collected all the different compounds in the sentence and considered them as multiple examples. The list of all the examples from outputs and references contained 6,025 items of the form: (sentence number, compound, position in sentence). If we considered for each sentence all possible compounds from various systems the list contained 2,245 unique compound-sentence number pairs from 806 sentences. That means we have 806 sentences from 1,002

where at least one system or the human translator produced a compound.

As discussed in Section 1.1.2 on compound merging strategies, the frequency of a compound, as well as its constituents, is substantial for its production. Although these merging strategies were mainly developed for SMT, they can also be related to NMT sequence-to-sequence algorithms namely to strategies for creating subword units. To determine the probability that the system created a particular compound, we compared the compound's frequency from the training data with the number of outputs that contained it (only for constrained systems that used the fixed training set). Due to a large number of examples of compounds (1306 for constrained systems), and the fact that the frequencies of contained compounds do not differ significantly, we grouped the observations according to frequency and averaged the number of systems that produced them.

Figure 3.2 shows the average counts of constrained WMT21 systems' outputs containing compounds depending on their frequency. The frequencies are displayed on a logarithmic scale. The dots in the plot represent the average number of systems that produced compounds for the frequency interval (placed at the mean value of the frequencies). The y error bars indicate the standard deviation in the count of systems for each group of compounds and the x error bars represent the frequency interval of the compounds. We can observe a growing trend in the graph: the compounds with higher frequency in the training data are more likely to appear in more systems' outputs. Nevertheless, the graph has some outliers, especially for low frequencies. This may be due to the limited number of found compounds with lower frequencies that influence the average.

The standard deviation in the count of systems containing particular compounds is quite large (see Figure 3.2). This may be due to the fact, that approximately half of the compounds appeared either in the output of only one system (263 out of 1,306) or in all of them (405 out of 1,306). However, the compounds exemplified in all outputs were more common for the more frequent words. The frequencies of compounds that appeared in all constrained translations were in the range of 5 (*Armeeschule* – army school) to 510,365 (*Flughafen* – airport), with an average of 44,475, while for the compounds contained only in one system output, the frequencies were from 19 (*Atemorgan* – breathing organ) to 427,165 (*Mitglied* – member), with an average of 18,753.

The statistic is also influenced by the possible synonyms that appeared in the translations but were counted as different words. Some examples are listed below:

(1) *Videoaufnahme* and *Videomaterial* (video recording)

(2) *Jugendgefängnis* and *Jugendstrafanstalt* (juvenile prison)

(3) *Sprühdose* and *Spraydose* (spray can)

We thought that these synonymous expressions could be the cause of a high number of compounds appearing in only one translation, but our hypothesis was proven wrong. For instance, the compound *Videoaufnahme* (Example 1) from sentence 6 appeared in four systems' outputs while its synonym *Videomaterial* appeared in four as well. Similarly, for the pair in Example 2 from sentence 16, five systems produced one of them and the other five systems produced the other. The synonyms in Example 3 had a distribution of three and seven, respectively.

In this section, we focused solely on compounds from GermaNet that were present in the training data and analysed their distribution. In the following

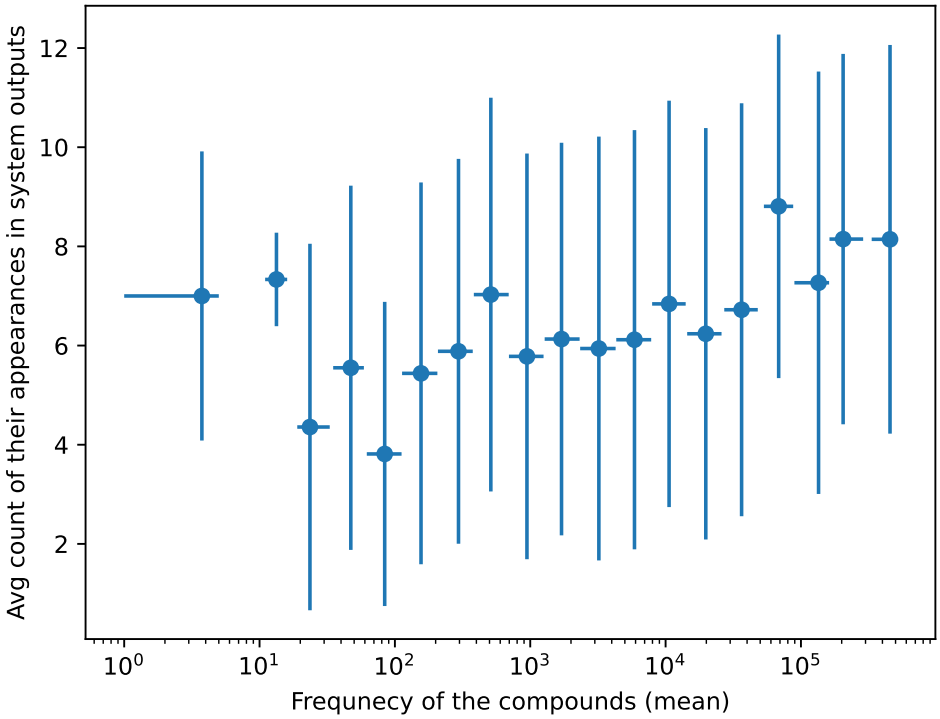


Figure 3.2: Frequencies of compounds in the training data and their appearance in constrained WMT21 systems' outputs

section, we will investigate whether the identified compounds were present in the training data or not. We will also attempt to identify more compounds in the translation, including those that are not listed in GermaNet.

3.3 Newly Created Compounds

MT models operating on subword units have the potential to generate unseen words in their output. We first examined the number of compounds from GermaNet that were produced by systems but were not present in the training data (marked as subsets II or III in Figure 3.1). This statistic depends on the lemmatization method used.

We found that there were no newly created compounds from GermaNet in the outputs of the constrained system. We expected this subset to be very small or empty, so it was not surprising. Some compounds were found by certain lemmatization methods but the manual analysis disproved them.

We also looked at whether there were any compounds from GermaNet that were not present in the training data. We found that the training data did not include approximately 3.5% to 5% (4,200 to 6,200 depending on the lemmatization method used) of the compounds from GermaNet. Therefore, there may be possible compounds that can be generated by the MT systems and identified by our methods, but we cannot be certain that they are relevant to the test text.

3.3.1 Words not Attested in the Training Data

We decided to explore a subset of compounds that were included in outputs of constrained systems but were not present in the training data or in the GermaNet list (i.e. subset I and the unlabelled remnants of the systems' outputs in Figure 3.1). However, there is no direct way to accomplish this. We collected all words that were not seen in the training data; note that we considered all words here, and manually verified which of them are compounds, see below. Table 3.3 displays the number of unseen tokens in the outputs and the counts of unique words that were not contained in the training data from the systems' outputs.

system	count of unseen tokens	unique count
C-eTranslation	146	106
C-HuaweiTSC	142	114
C-UF	139	112
C-Manifold	134	112
C-WeChat-AI	128	105
C-ICL	127	106
C-UEdin	123	103
C-nuclear_trans	122	100
C-Nemo	121	96
C-BUPT_rush	116	98
C-P3AI	114	97

Table 3.3: Statistics of words not seen in the training data for constrained systems

Determining whether a word that appears to be a compound is an existing or conceivable German word is not easy. We can consider all compounds produced by native speakers as proper German words. To identify novel words, we searched large monolingual corpora, such as Araneum Germanicum Maius [Benko, 2014] or the DWDS dictionary [Klein and Geyken, 2010]. To include compounds used in German articles or web pages, we used Google search.

We conducted a manual analysis using these resources to determine whether the words produced by the systems exist. The systems produced a total of 304 unique new words that started with a capital letter, indicating that they were possible nouns. Approximately half of them were found by Google. During the analysis, we discovered various groups of words. Some words were of foreign origin, such as the English verb *maced* (capitalized because it was so in the source text), human names like *Shaquia* and *Bhadauria*, and geographic locations like *Mambourin*. Regarding compounds, we discovered an example of a joint English phrase, *Speakupfordemocracy*, and many German compounds. Out of 304 novel nouns, we manually determined 229 of them as compounds. The exact number of identified compounds and foreign words for each constrained system is displayed in Table 3.5 below.

We examined the German compounds and discovered many words that appeared to be compounds and were made up of meaningful constituents but were neither included in the training corpus nor found by Google. Naturally, they were also not found in DWDS. Example 4 lists several instances of this phenomenon.

Most of the examples make sense as two separate words and combining them into a compound is possible. For instance, Example 4d *Quarantäneentscheidung* can be split as *Quarantäne* (quarantine) and *Entscheidung* (decision). We also provide examples of more complex words produced by the systems that do not have any known sense (see Example 5). Their two constituents can form proper German words (Examples 5d and 5e), but their concatenation is not known as a German compound. However, there are also examples that could not be clearly divided into two parts that make sense (for instance, 5b or 5c were formed from three such parts).

The systems also produced compounds that existed and were found by Google but were not contained in any mentioned corpora. The examples of these rare words we found during the analysis are listed in Example 6. These words were also produced by humans in some texts or articles but did not belong to a common vocabulary. In total, 103 of 229 novel compounds were found by Google. This analysis provides several examples of the productivity of NMT models in terms of compounds. We examined these examples further and searched for them in a bigger German corpus, namely in *Deutsche Referenzkorpus* (DeReKo).[1] The DeReKo corpus revealed that beside all compounds from Example 6 Examples 4b and 4d can aslo be considered as existing compounds.

(4) Novel words made from meaningful constituents
   a. *Kirchenkanister*
   b. *Kondolenzbotschaft*
   c. *Gladiatorenmodus*
   d. *Quarantäneentscheidung*

(5) Very complex unknown words made from meaningful constituents
   a. *Sanktionsüberwachungsteam*
   b. *Gefangenenfreistellungsprogramm*
   c. *Passagierlokalisierungsformular*
   d. *Notfallgesundheitsdirektorin*
   e. *Telekommunikationsnetzausrüstung*

(6) Very rare compounds
   a. *Flughafenvertrag*
   b. *Pandemiekrise*
   c. *Kartoffelwurzeln*
   d. *Schlüsselarbeiterin*
   e. *Republikanerkollege*
   f. *Amateurfehler*

After discovering many newly produced compounds in the systems' outputs, we also explored words produced by human translators in the references that were not contained in the training data, in order to compare them. As the organizers of the WMT21 found out, reference B is a post-edited output of one of the participating MT systems. We also found a lot of newly created words in its output that suggest the text was produced by a machine. However, during

[1] https://www.ids-mannheim.de/digspra/kl/projekte/korpora

our manual analysis, we also found several words in the reference translations that were not included in the training data. We can assume that these words were created correctly and reflect the discourse situation of the source test text. Particular phrases in the source text encouraged the translators to create these compounds.

We collected all the words produced by humans in the reference translations of the test text. We are aware of the fact that comparing the vocabulary of human translations to training corpora might not be ideal to demonstrate productivity regarding composition. However, we can consider the huge training corpora as a sample of common vocabulary knowledge. The counts of the tokens from human translations that were not present in the training data are listed in Table 3.4. Reference D had the most new words, as it also had the highest number of compounds from the GermaNet list in its translation. The number of newly created words is similar for human translations as it was for MT outputs (see Table 3.3). All the references together contained 193 unique novel nouns of which 82 were also present in the MT outputs.

system	**count of unseen tokens**	**unique count**
ref-D	144	110
ref-B	132	100
ref-A	129	100
ref-C	122	93

Table 3.4: Statistics of words not contained in the training data for reference translations

We detected several novel compounds from our examples also in the reference translations: The compounds *Kondolenzbotshaft* and *Gladiatorenmodus* were found in references B and D, while references A and C contained a modification of the second compound, *Gladiatormodus*. Two of the complex compounds that seemed to have no sense were also created by humans, namely the word *Sanktionsüberwachungsteam* in references B and C and *Passagierlokalisierungsformular* in references A, B, and C. We found three of the listed rare compounds in the references – *Flughafenvertrag* (in references A, C, and D), *Pandemiekrise* (in references B and C) and *Kartoffelwurzeln* (in all references). That affirmed that MT systems are capable of producing unseen but meaningful compounds. Half (150 out of 304) of the MT-generated novel nouns were found to be existing words (based on google search) and less than a third of them (82 out of 304) were approved by reference translations. We can be certain that the novel nouns that were also present in the reference translations were created correctly according to the context and information need in the test text, but we can not easily decide the correctness of the other novel words.

After providing manual analysis and listing some examples, we grouped the observations together. Table 3.5 displays the number of novel nouns created by constrained MT systems, their cooccuracne with reference translations and their distribution into categories. We distinguished three categories: compounds, foreign words or names, and other, such as web domain names or meaningless words. Only the first two categories are listed in the table. We also estimated

system	# nouns	n. in ref	# comp.	c. in ref	# foreign
C-Manifold	106	52	69	22	34
C-HuaweiTSC	102	57	58	24	36
C-UF	101	58	60	24	36
C-WeChat-AI	95	54	51	19	35
C-UEdin	93	56	49	20	37
C-eTranslation	92	56	55	23	32
C-Nemo	87	51	44	17	38
C-nuclear_trans	87	47	44	13	35
C-P3AI	86	45	49	15	32
C-ICL	82	47	41	15	35
C-BUPT_rush	81	43	40	11	34

Table 3.5: Categories of unseen words produced by constrained systems according to manual analysis

how many of the novel compounds were also present in the reference translations. In most of the constrained systems, more than a half of novel nouns appeared to be compounds, as shown in Table 3.5.

To conclude, the MT systems are, same as humans, capable of generating novel words although it did not seem so when relying on a fixed list of compounds. At the same time, the number of compounds in the translations is still higher for human translators than for the MT systems when we count both novel words and compounds found by GermaNet.

4. Training Own Transformer Model

As seen in Chapter 3, human translators produce more compounds than state-of-the-art MT systems. The goal of the thesis was not only to analyse the appearance of compounds in translations but also to determine the conditions that may lead to producing them. To achieve this, we trained our own Transformer model for English-German translation on a fixed dataset, namely on parallel corpora provided by WMT21 (see Figure 2.3).

4.1 Preprocessing of Data

For further analysis of compound productivity, we trained several variations of the Transformer model. The modifications mainly concerned the preprocessing of the training data. We preprocessed the training data using a script inspired by FAIRSEQ examples for translation.[1] We adapted the script and performed different preprocessing variants. The segmentation to subword units was provided by the SubwordNMT implementation of BPE (see Section 2.2.4). After applying the BPE encoding, the data was binarized using the FAIRSEQ *preprocess* script. We experimented with the size of the joined vocabulary for subword units and utilized variants with two different dictionaries for English and German. The first model had a joined dictionary size of 40k and was preprocessed with the default random seed set to 1. In the second setup, we utilized two separate vocabularies with size 40k, and the seed was also left to default. We experimented with a much smaller vocabulary in the third setup to see whether it could help with the production of the compounds. For the fourth model, we set the seed to 1,000 to examine how are the results biased by the random seed. For clarity, all the versions are listed in Table 4.1.

system	**seed**	**type of dictionary**	**size of dictionary**
T40k	1	joint	40,000
T2x40k	1	separated	2 x 40,000
T10k	1	joint	10,000
T2x40k-2	1,000	separated	2 x 40,000

Table 4.1: Training setups of our Transformer model

4.2 Training Setup

The models were trained using the FAIRSEQ framework as described in Section 2.2.1. We trained the models using the default FAIRSEQ Transformer configuration containing 6 decoder and 6 encoder layers, each with eight-headed attention. The setup differed from the default configuration in the following ways.

[1] https://github.com/facebookresearch/fairseq/tree/main/examples/translation

The parameters were inspired by EdinSaar's submission to WMT21 [Tchistiakova et al., 2021]. We operated on batches of a maximum size of 4,096 tokens. We used the Adam optimizer with setting $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e-9$. The dropout was set to 0.01. We utilized the GELU activation function. The learning rate was set to $3e-4$ and scheduled with an *inverse_sqrt* scheduler. We set 16,000 warmup updates with an initial learning rate of $1e-7$. The criterion for training was label-smoothed cross-entropy. The models were trained on a heterogeneous grid server that contains Quadro RTX 5000, GeForce GTX 1080 Ti, RTX A4000, and GeForce RTX 3090 cards. We utilized 8 GPU cards across several weeks to train the models. We did not train all versions of the model at once and the training was not continuous, so we could not specify the exact time of training.

The training was stopped after at least 13 epochs for each model. The validation was done every 5,000 updates, and the checkpoints were saved afterwards. We kept only the checkpoints for each epoch to save disk space. We let the training of the first model run longer than the other variants, so we could explore its features in later phases of training. It is further described in Chapter 5.

4.3 Overall Translation Quality

We observed the overall translation quality of the models after each epoch of training. The BLEU scores computed by FAIRSEQ against reference translation A during the training of all four models are displayed in Figure 4.1. As we used the SacreBLEU implementation to evaluate the overall quality of translations in the thesis, we recomputed the scores with this implementation. The BLEU scores against reference A and against all three references are shown in Figure 4.2. The results from both implementations of BLEU did not differ significantly, although the scores from the FAIRSEQ implementation were lower ranging from 16 to 19, while the SacreBLEU scores ranged from 20 to 23 for reference A.

The model with the small subword dictionary had the lowest scores, especially in the early phases of training, as displayed in Figure 4.1 and Figure 4.2a. The other three models did not differ much in scores. The variants of the model with separate dictionaries for the source and target languages achieved higher scores than other variants at the beginning of training, but the scores for these variants did not improve much in later epochs. The second model had a peak in score after the seventh and tenth epochs. The first model with a joint vocabulary outperformed the other variants in all phases of training except for the beginning, seventh, and tenth epochs where the second model achieved a better score.

The scores were more balanced for the different variants of the models when using all references for evaluation, as shown in Figure 4.2b. They ranged from 37 to 43. Similarly to Figure 4.2a, we observed that the systems with two separate vocabularies achieve higher scores after the first epoch of training than the systems with one joint dictionary. Contradicting the evaluation using only reference A, the second model outperformed the first model in most epochs when using all three references for similarity.

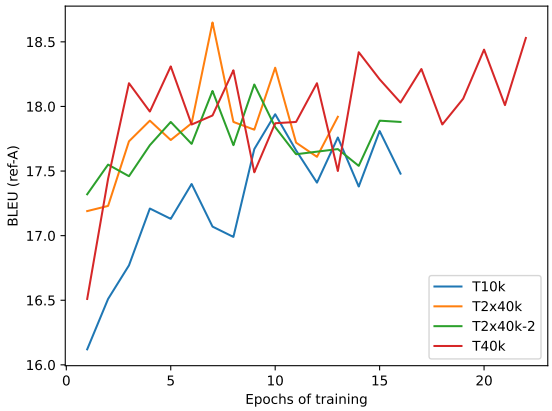
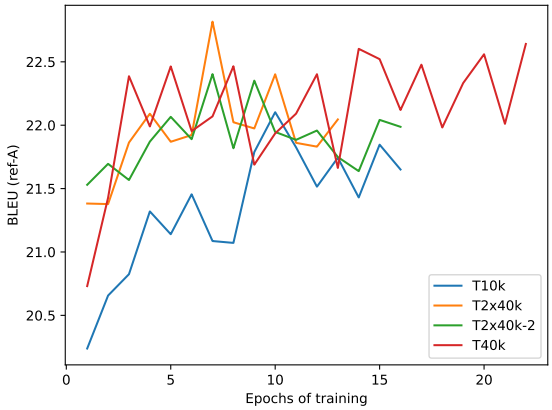
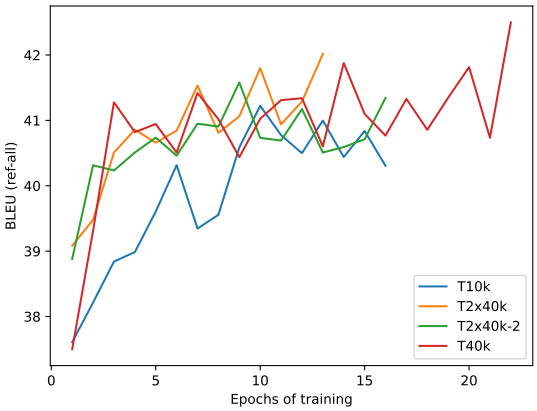


Figure 4.1: BLEU score for reference A during training of all four versions of the Transformer model computed by FAIRSEQ



(a) BLEU score for reference A



(b) BLEU score for all references

Figure 4.2: BLEU scores during training of all four versions of the Transformer model computed by SacreBLEU

5. Compounds in Our Transformer Model

In Chapter 3, we observed that humans produce many more German compounds than MT systems. By adjusting the models to generate more compounds in their outputs, we may be able to improve MT quality. In order to analyse the behaviour of Transformer models and figure out the conditions that lead to the production of compounds, we trained our own models as described in Chapter 4. This chapter provides a detailed analysis of compounds in our model outputs.

First, we studied the outputs of the systems and compared them to human references and state-of-the-art systems from WMT21, as presented in Section 5.1. Then, in Section 5.2, we analysed further aspects that we thought could affect the generation of compounds in the outputs of the systems. We also explored the models' features and scores during output generation. We provided examples and an evaluation of forced decoding in which we passed prefixes of different lengths to the model and observed how the outputs differed. We focused mainly on compound production and how the prefix influenced it. This analysis is presented in Section 5.4.

5.1 Analysis of Compounds Appearance

In Chapter 4, we presented the general translation scores for all four variants of the Transformer model. In this section, we shift our focus to quality concerning the counts of produced compounds. We counted sentences containing compounds and also all occurrences of compounds in the text. We provide the results for all models' epochs compared to the reference translations. Observing the counts after each epoch may lead to an understanding of how the production of compounds evolves during training. The counts during training are displayed in Figure 5.1.

The counts of compounds were not generally just growing during training. We can observe peaks and low points on the curves (see Figure 5.1). The first version of the model, trained on the joint dictionary of size 40k and for more epochs than the other variants, achieved the highest number of compounds in its output. However, the second variant using separate vocabularies for the languages produced more compounds than the first setup at the beginning of training. We left the training of the first model to run longer because the production of compounds seemed to grow. Nevertheless, the growth was only for a few epochs after epoch 15, and then the count of compounds in the outputs started to decline.

The model with a smaller vocabulary that was forced to split more words into subword units did not show any improvements regarding compound production when evaluated against GermaNet. The model performed even worse than the previous models, as shown in Figure 5.1. We expected that the splitting of more words would lead to some improvements. Nevertheless, it might not be a problem of the size of the dictionary but the algorithm for splitting. It split words according to their frequency in the training data and did not consider the morphological features of complex words.

Utilizing a different seed for the preprocessing did not result in significant

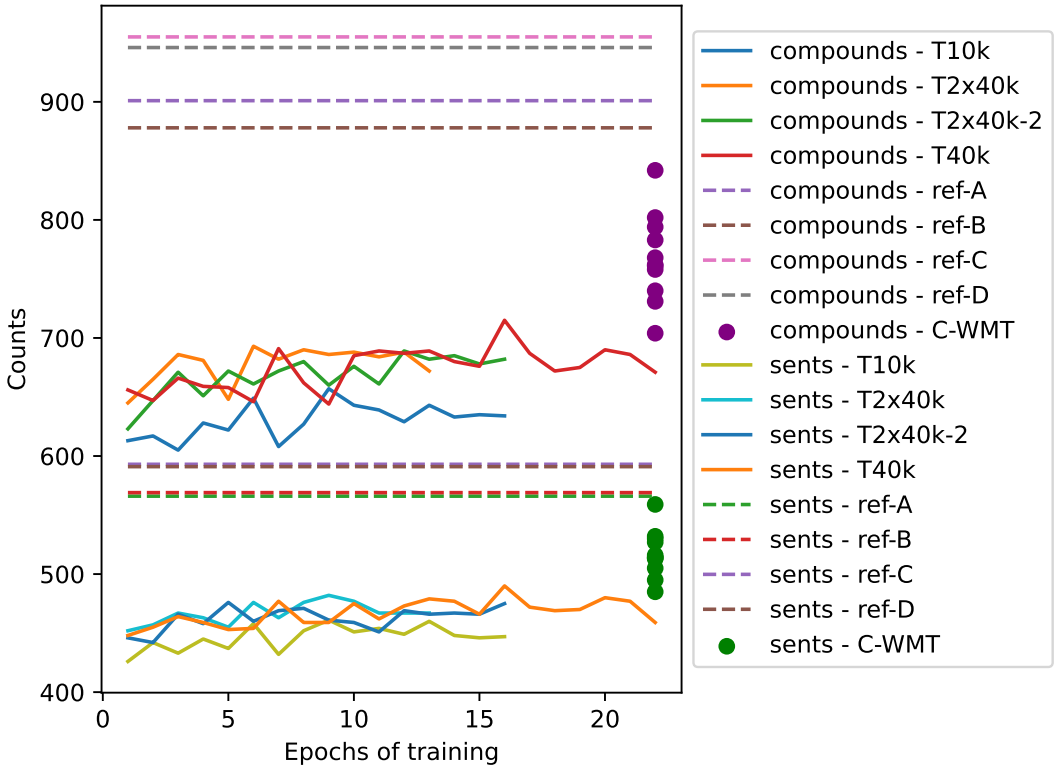


Figure 5.1: Counts of compounds in the system's output during the training of our Transformer compared to references and WMT21 systems

changes in the counts of compounds. The number of sentences containing at least one compound did not differ much between the setup with joint vocabulary and the setup with separate vocabularies of the same size, as well as the modification with a different preprocessing seed. The counts of sentences with compounds were slightly lower for the third setup with a smaller vocabulary.

We compared the numbers of compounds and sentences containing them to references and constrained WMT21 systems. As shown in Figure 5.1, both the counts of compounds and sentences were much lower for all of our systems compared to human reference translations. Although there are some differences among the references regarding compound production, the results are incomparable to our system outputs.

We visualised the number of compounds and sentences containing them for the constrained WMT21 systems with dots in the graph. The exact counts for each system were displayed in Table 3.1. The values for WMT21 systems fill the space between our models and human translations. Our best-performing systems reached comparable values to the worst WM21 systems for both counts: compounds and sentences. It is not surprising that our vanilla Transformer model did not outperform the state-of-the-art systems. These differences could also be observed in the overall MT quality of our Transformer models compared to WMT21 MT systems. As shown in Table 3.2 and Figure 4.2, our best systems achieved BLEU 23 and the WMT21 systems' quality was in the range of 24 to 31 BLEU (measured on reference A).

As described in Section 3.1, there was a correlation between the number of

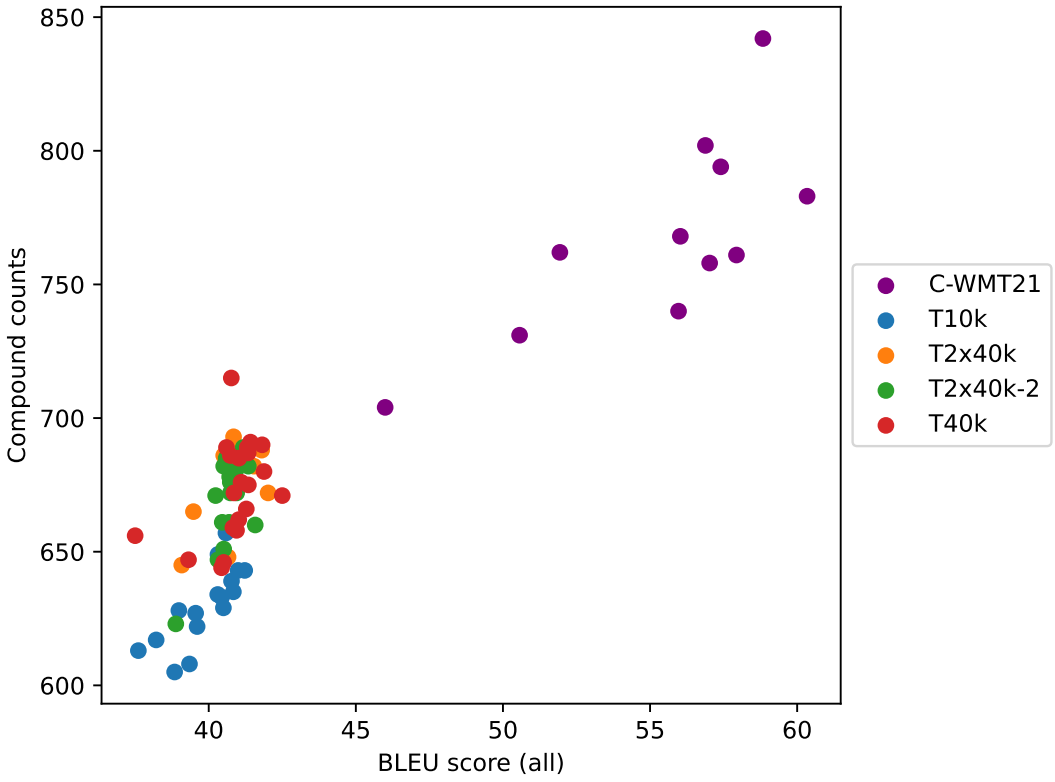


Figure 5.2: Comparison of overall translation quality to the number of produced compounds for constrained systems an our systems

generated compounds and the BLEU score of WMT21 translations. We visualised the relation of both scores for all the constrained MT systems, including four versions of our Transformer, as shown in Figure 5.2. The scores did not differ much for our Transformer models, as observed earlier. However, the graph confirmed the correlation between the overall quality of translations measured by BLEU and the number of generated compounds. Their dependency is almost linear. This implies that we could possibly rely on overall quality when dealing with compound production in constrained systems.

5.2 Aspects Affecting Compounds Generation

We already mentioned several conditions that can affect compound generation in Transformer models during the training and analysis of our models. We experimented with the size of vocabulary in training, as described in Chapter 4. In Section 5.1, we observed counts of compounds during the training of all versions of our Transformer models.

The hypothesis that reducing the size of subword vocabulary would force the model to be more creative and generate more compounds in the output was not supported by GermaNet's list of compounds, as shown in Figure 5.1 when comparing T10k with our other models. However, as discussed in Section 3.3, the ability to generate new compounds cannot be measured by a fixed list of compounds.

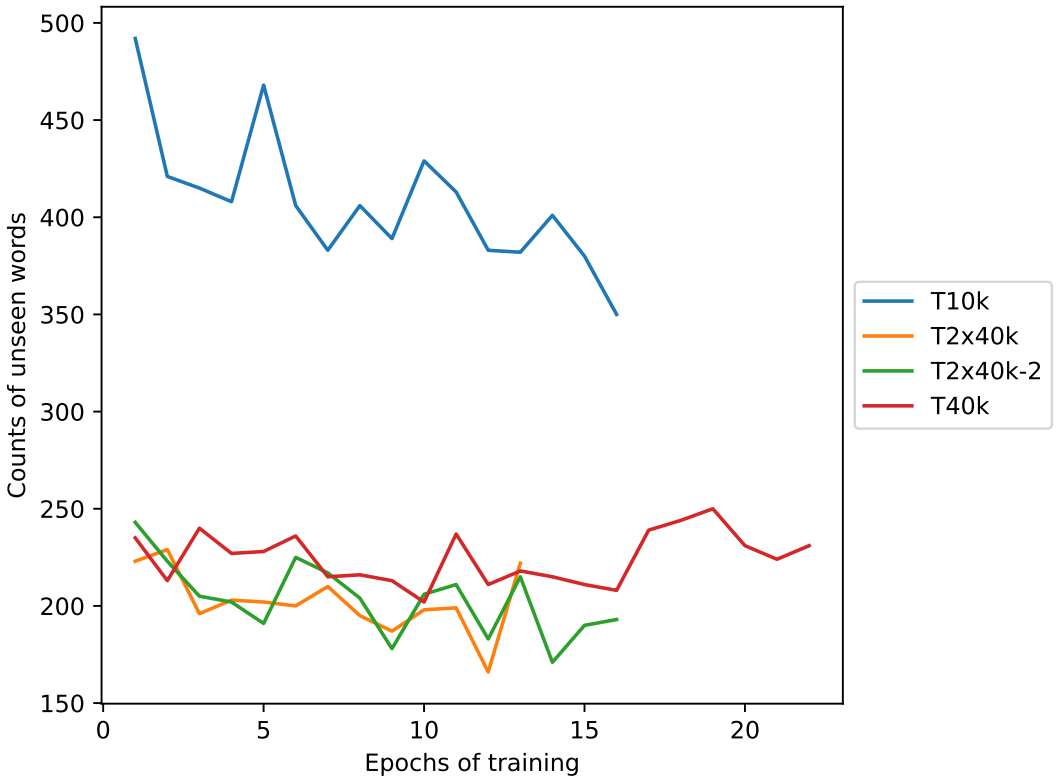


Figure 5.3: Counts of words from our Transformer outputs that were not included in the training data

5.2.1 Novel Words

To further investigate the creation of novel words, we collected the words in the translations that were not present in the training data. These words could potentially be compounds, as seen in Section 3.3 during the manual analysis of unseen words in WMT21 translations. Figure 5.3 displays the number of unseen words in our Transformer models for each epoch of training. The third version of our model, which had a smaller vocabulary than the other models, generated many more new words. After the first epoch, the system produced 492 words that were not present in the training data, of which 466 were unique. The count of newly produced words decreased in the later phases of training and after the sixteenth epoch, only 350 new words were created.

We manually analysed the words produced by the third model. The list of unique words from the outputs of all different epochs of the model contained 2,024 novel words. Of these, 1,320 were considered nouns because they started with capital letters, which is a sign of nouns in German. The shortest word consisted of only three letters, while the longest contained 48. The average character length of the newly created words was 12. We expected the longer words to be compounds. Unfortunately, the longest word (Example 7a) did not make any sense in German since it contained 43 "i" letters after each other. The second-longest word was at least made up of meaningful components (Example 7b).

(7) The longest novel words

a. *Besiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii*

b. *Bankmitgliedertiruchirappalli*

During the analysis, we observed that the generated words mostly combined subwords from different words. The combined subwords did not make sense together in German, as presented in Example 9. Some compounds, such as Example 9c (prisoner exchange programme), were built from meaningful stems but the created words did not have any meaning in German yet (there is probably no exchange programme for prisoners, but both "prisoners" and "exchange programme" are meaningful constituents and could potentially create a new word together). The other group of words contained compounds generated from subwords from different words, as shown in Examples 9a and 9b, or by repeating subwords from the same word, as in Example 9d.

We also noticed that the longer the produced words were, the less probable they were to make sense. However, we found some long words that were meaningful in German, as Example 8b. The other meaningful words were mostly technical terms (Examples 8a and 8c).

(8) Existing words
   a. *Säuglingsmedizin*
   b. *Coronkoviruspräventionsmassnahmen*
   c. *Wärmeverletzung*
   d. *Kriminalitätsexperten*

(9) Non-existing words
   a. *Beschäftigungspersonstigen*
   b. *Befürwortwörtlich*
   c. *Gefangenenenaustauschprogramm*
   d. *Schaufschaufschaufschaufeln*

The third system generated an extraordinary number of newly created words; however, as we discovered during our analysis, most of them were not real German words. Although our hypothesis about reducing the vocabulary size was not completely wrong and the model produced a big number of new words,it generated the least compounds from GermaNet. Moreover, most of the newly created words were not meaningful German compounds.

5.2.2 Compound Production vs. Frequency

As observed by the WMT21 MT systems (shown in Figure 3.2), the more frequent compounds were more likely to be generated than those with lower frequencies in the training data. We decided to explore this phenomenon in our models as well. We focused our analysis on the models after each epoch of the first variant of our Transformer. In Figure 5.4 we display the average number of systems that generated compounds of particular frequencies. Dots on the graph represent mean values on both axes. Horizontal lines present intervals of frequencies, while vertical lines display the standard deviation from the mean values.

As shown in Figure 5.4, only a few models of our Transformer produced compounds with frequencies under 10. Then, the average number of systems that generated compounds increased as the frequency grew up to 10^5. There were

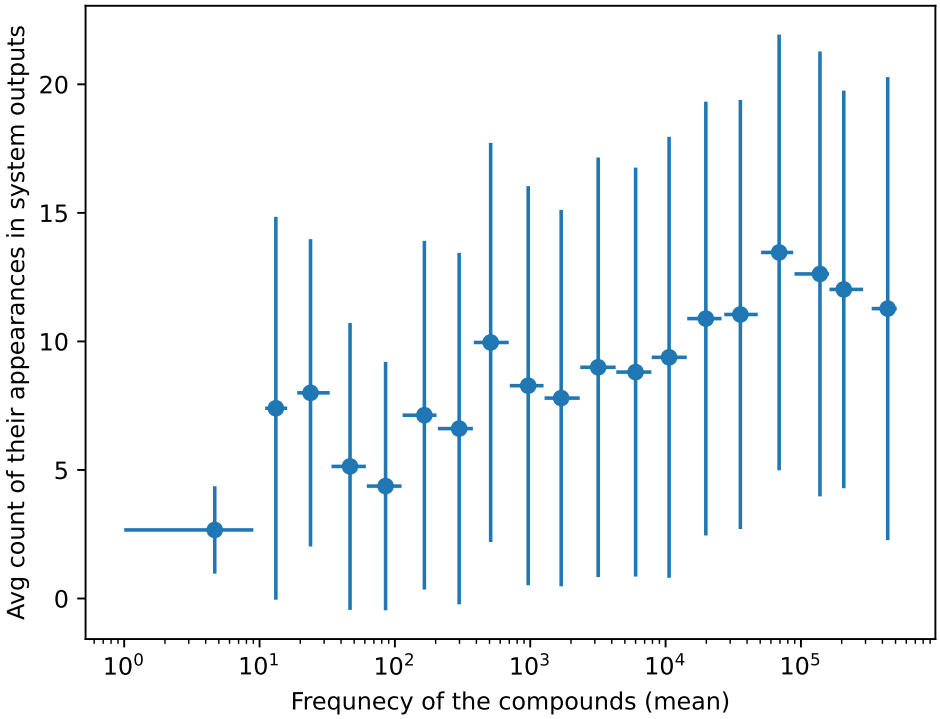


Figure 5.4: Frequencies of compounds in the training data and their appearance in outputs of first modification of our Transformer for all epochs

probably only a few compounds with frequencies above 10^5, so the number of systems producing them decreased after this frequency. There are only a few outliers from the growth curve in the graph.

Similar to Figure 3.2 for WMT21 systems, we observed a growing trend in the graph (Figure 5.4). The trend was even more visible for our Transformer than it was for the WMT21 systems. Since each of the systems submitted to the WMT21 conference focused on different phenomena in the language, the differences in compound production were not significant on average regarding their frequency in the training data. In our setup, we utilized just a vanilla Transformer and did not specialise in improving the MT quality in any way.

5.3 Compounds as Inference Shortcuts

Furthermore, we proposed a modification to the model that could increase the number of produced compounds. Our assumption for this modification was that the path through the search space is easier for words that consist of more subwords, like compounds. When the model chooses that path, the next steps become much easier because the choice of the right subword in the middle of the word is straightforward. An example of a beam search space is displayed in Appendix A.2.

We suggested computing perplexity in each decoding step and including it in the total score of that node in the search space. The perplexity should be computed for a few following decisions, for example, three. This approach would

add a look-ahead to the model and prefer sentences that contain more complex words. We tested our proposed modification on the outputs from the beam search. We chose a sentence where we knew that the model generated a sentence without a compound, but the compound was listed in the concerned hypothesis in the beam search. The perplexity of the node with the compound start was really lower than the perplexity of the other possible node.

In the end, we did not implement this modification in the FAIRSEQ generation step because of the lack of time and the complexity of the FAIRSEQ implementation. However, it could be investigated in future work. We believe that it could lead to better results regarding the number of generated compounds because this way of thinking seems more similar to that of humans. People do not maximise the probability of the sentence; instead, they think more in an economic way. An effort of human speakers is to express their thoughts as clearly and easily as possible.

5.4 Forced Decoding Using Prefixes

To investigate further properties of models that influence compound production, we explored the behaviour of the models when we gave them the first words (prefixes) of the translated sentences. The models were forced to use these words as beginnings and then continue translating the source sentences. This process of influencing the outputs is called forced decoding. We observed how the outputs would look if we changed the early steps of the inference. We fed prefixes of different lengths to the models and then compared the outputs. The analysis focused on the number of produced compounds.

5.4.1 Hinting towards More Compounds?

We ran experiments on the first setup of our Transformer and forced the models to use the first n words from the given reference (reference A) for the translation. We call the n prefix length. We generated all the outputs with prefix lengths from 0 to 40. The counts of collected compounds from GermaNet are displayed in Figure 5.5. The number of found compounds increased with the growing length of the prefix and reached almost the value from reference B for prefixes of length above 30 words. However, reference B had the fewest compounds and was a post-edited version of MT output, while the other references that were truly translated by humans included more compounds. Although it seemed that we could influence the model by hinting it the first words to generate more compounds in the translations, this statistic was not accurate because it also counted the compounds hinted by the reference. Moreover, the number of sentences that the model could influence decreased with the prefix length. The short sentences were just copied from the reference and therefore brought inaccuracy to the graph.

To overcome these two errors in the statistic, we stripped the prefix of the particular length from each sentence before collecting compounds from it and displayed the relative number of compounds, not the absolute. Figure 5.7 shows the percentage of possible compounds from reference A that the systems generated. We counted only the compounds that were not included in the prefix, so the number of concerned sentences decreased with the growing prefix length. We

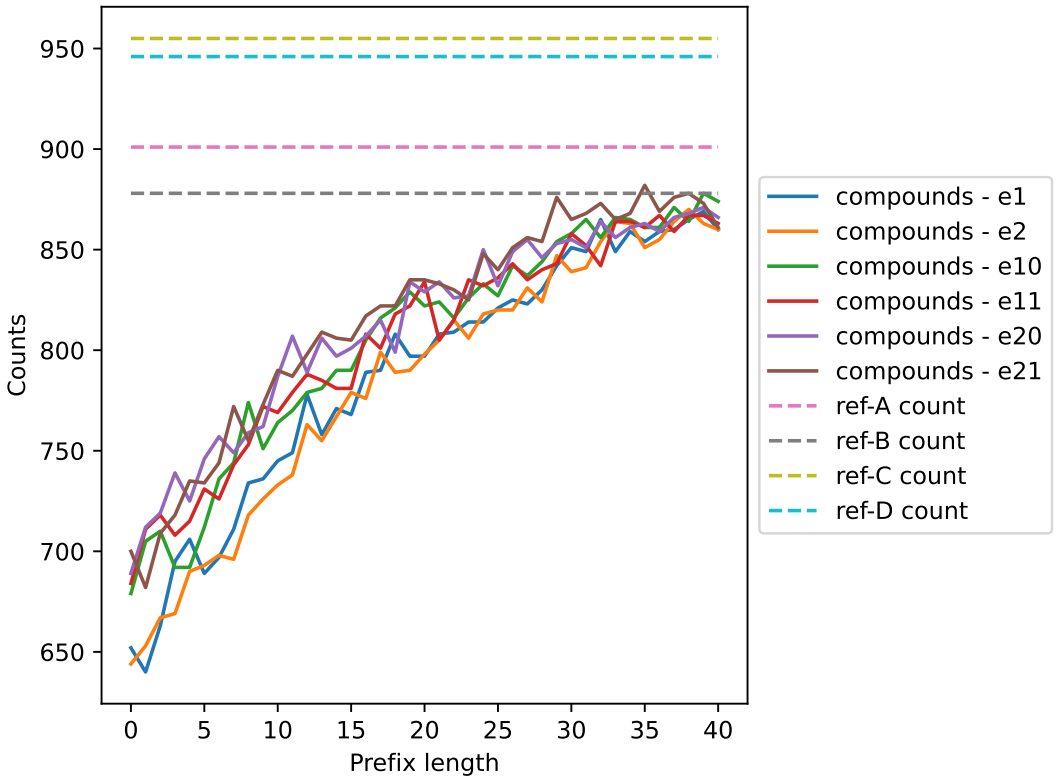


Figure 5.5: Counts of compounds in our Transformer depending on prefix length passed to the system

visualised the number of compounds in reference A that were not contained in the prefix, as displayed in Figure 5.6. It was taken as a basis for the search for compounds in the outputs of our Transformer. As shown in Figure 5.6, the number decreased almost linearly with the prefix length. It started with the value of 901 compounds for the whole sentences and descended to 60 for the prefix length of 40.

Figure 5.7 shows that the proportion of produced compounds from reference A did not change much for different phases of training. The curves for all displayed epochs have a similar shape. Nevertheless, the Transformer generated fewer compounds in the early phases of training, which is especially notable for prefixes under 20. For prefixes shorter than 20 words, the percentage of produced compounds was around 40% after one or two epochs and around 45% after more epochs of training. The peak of generated compounds was for prefixes of lengths 28 to 30, where all of the systems generated the highest rates of compounds. The number of possible compounds from reference A in sentences after the 28th word was 141, and the systems produced 49%–63% of them. The percentage of compounds descended rapidly for prefixes of length above 30. The accuracy in generated compounds then increased again for the longer prefixes, but it is not significant since the number of possible compounds was very low (under 100 compounds, as shown in Figure 5.6).

The peaks in the graph may be caused by different frequencies of the compounds in the training data. As we have seen in Section 5.2, the more frequent compounds are more likely to be generated. And since the distribution of com-

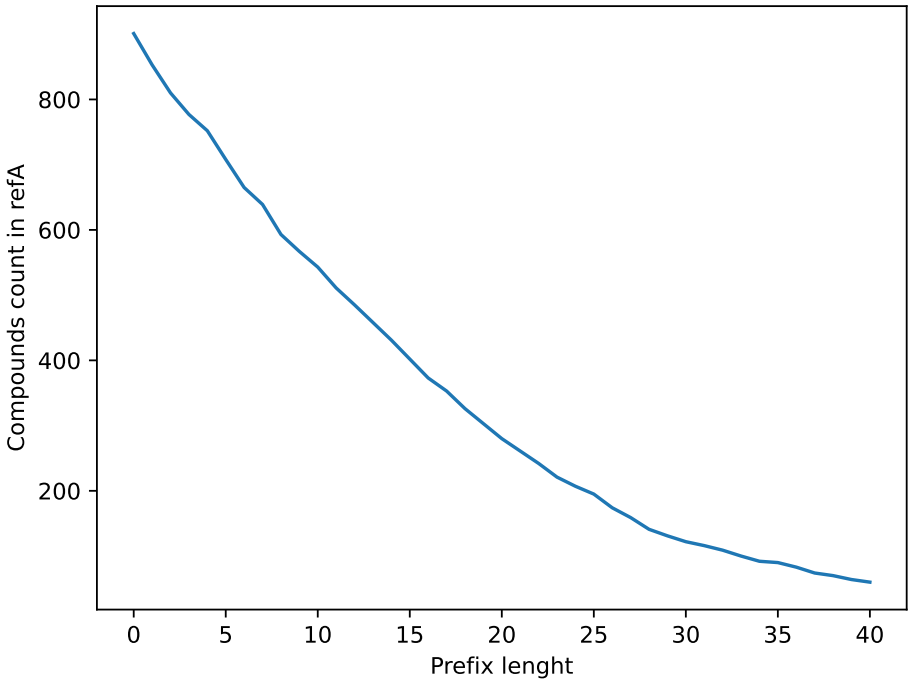


Figure 5.6: Count compounds in ref A depending on prefix lenght

pounds in the reference is not balanced, some compounds at the end of sentences could be more frequent in the training data than the compounds at the beginning and were therefore generated more times in the outputs.

5.4.2 Just One Word Hint Sufficient?

To investigate the impact of prefixes on the generation of compounds, we collected prefix lengths that led to the generation of each particular compound from a list taken from reference A. We utilized this statistic on the first setup of our Transformer for models after each epoch and noticed that in many cases, the generation of a compound was influenced just by hinting the first word to the model. It often happened that the system did not originally generate the particular compound, but when we gave it a prefix of at least one word it produced it. So, the compound generation was influenced just by the first word of the candidate translation.

This led us to aggregate the statistic and count all compounds that were generated by hinting just one word and remained in the outputs for longer prefixes (called "first prefix"). We also summed all cases where it happened with a later prefix than one (called "later prefix"), as well as the compounds that were generated when we hinted all preceding words (called "preceding prefix"). For the first two cases, we provide a soft statistic where at least 80% of the prefixes up to 40 or to the position of the compound led to its generation, as well as the strict statistic where all of the prefixes had to lead to the compound. The collected counts for the system after the 22nd epoch are displayed in Table 5.1. We found that 24% of compounds that were not present in the original translation of the

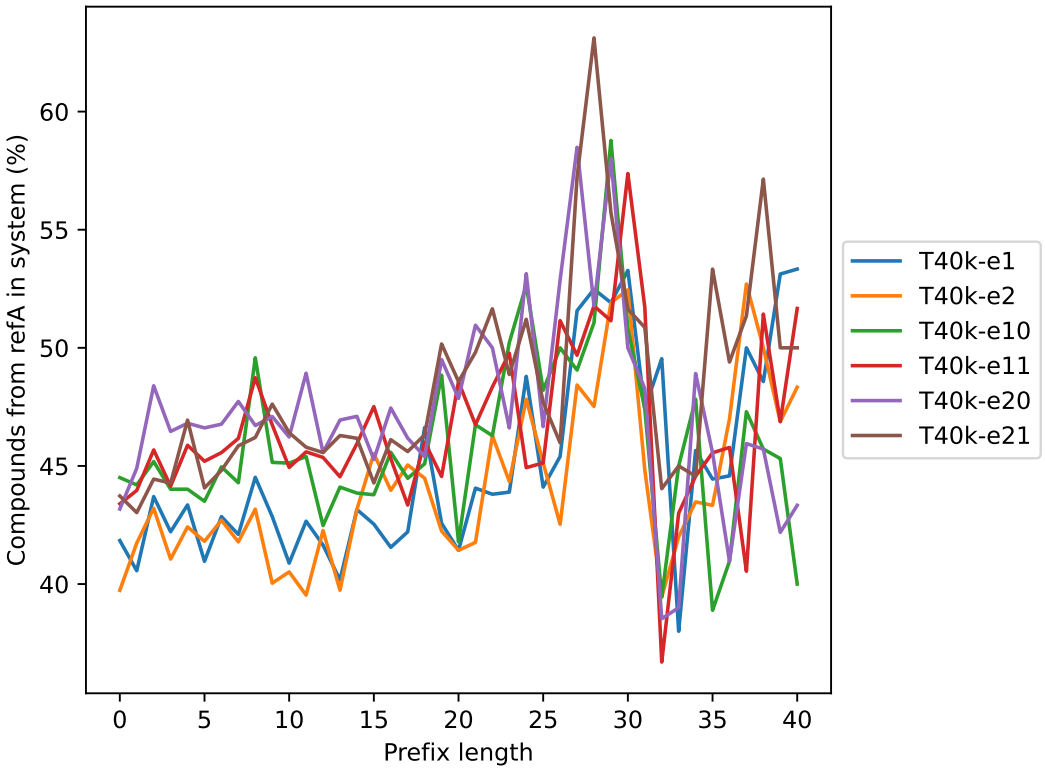


Figure 5.7: Percentage of compounds from ref A comprised in Transformer outputs with fixed prefixes

model appeared in the outputs where we hinted at least one word. In another 17% of cases, it happened after some later prefix. So, 41% of compounds were suddenly created after limiting the search space during output generation.

To further investigate this phenomenon, we conducted a manual analysis of the outputs. We found that sometimes the first word influenced the system to produce a more German variant of a word, such as *Gleichgewicht* instead of *Ballance* or *Flughafen* instead of *Airport*. We believe that the first word changed the search space of the sequence-to-sequence algorithm, leading to the generation of a different synonym for the particular word. We also observed that the influenced system could produce a compound in a different position than in the reference translation and the generation was also affected by the prefix. For example, a prefix of length 5 was long enough to change the wording of the sentence and caused the model to produce the compound, but earlier than in the reference (position 9). Therefore, the compound was not counted by longer prefixes (6 and longer) and was not added to the "later prefix" case.

We also discovered examples where the first hinted word was the same as the first word in the original output of the model, but the prefix still changed the output. This fact indicates some errors in the search algorithm. The types of possible errors are described in Section 1.2.1.

We visualised the search space of the beam search for one concrete sentence where the original model did not create a compound but the models influenced by prefixes did. We chose a sentence with the number 437 generated by the model after the 22nd epoch of the first setup of our Transformer. The original

variant of change	number of cases	from them strict
first prefix	26	12
later prefix	19	9
preceding prefix	6	
other	58	

Table 5.1: Agregated statistic on changes of appearance of compounds in the output based on prefix

translated sentence started with the word *Herr* (Mister) and so did the sentence in reference translation. The observed compound *Kinderspiel* (children's game) was at the end of the sentence. However, it was generated only by the influenced models and the original model produced instead a word that did not make any sense in German (*Klacks*). We attached the sample of the search process for the original model (see Appendix A.2) and for the model influenced by hinting the first word (see Appendix A.3). The influenced model developed only hypotheses beginning with the hinted word, all other beginnings were scored minus infinity. This fact influenced the shape of the search tree. The search tree of the influenced model developed more on one side (left in our figure), while the original model elaborated all of the first five hypotheses and was, therefore, more balanced.

We discovered that these two outputs contained the same path through the search space to step 15 where their paths split based on the variation of the score for the next word. Since the difference in their paths was only in the score of the next word, we tried to explain it. However, there was some non-determinism in the model which we did not manage to find in the complex implementation of FAIRSEQ. This path was in both cases the best hypothesis and resulted in the final translation hypothesis. The sentence in the original model was made from 39 subwords while the influenced output sentence was from only 36 subwords. The original model decided to repeat some words in the sentence and in the end produced another word instead of the compound, otherwise the translated sentences were the same. Even though only the influenced model resulted in the generation of the compound, both models considered variants of the sentence with or without the compound. The decision on whether to produce a compound was made in the last step based on the final scores of the hypotheses. The final hypothesis of the influenced model scored better than the one from the original model. So we discovered a search error in this particular sentence.

As stated before, the hinting of the first words changed the search space of the model. It was shifted to one side and had a chance to explore more hypotheses starting with given words. We have seen previously in this section that it did not bring any significant improvements in terms of the compound production of the models. Although we found a lot of examples where hinting at the first word helped, we could not systematically influence the models to generate more compounds.

5.4.3 Discussion

Why humans produce more compounds in their texts remains an open question. As Holtzman et al. [2019] found out, models using beam search and other maximization-based decoding methods tend to a generation of very probable output and do not share diversity in vocabulary with human texts. This claim goes along with our findings. Compounds are generated based on their frequency, and shifting the model to another subspace of the search space did not change this behaviour. We suggest that one of the possible reasons for human translators and speakers to produce more compounds might be the language economy. Using compounds instead of multi-word expressions shortens the output sentences, which people tend to make brief and clear when speaking. The use of compounds can be helpful for that purpose because multi-word expressions often contain other words like prepositions and articles that are not needed in compounds. Additionally, compounds do not require the same level of exactness as multi-word expressions with prepositions do.

To compare the overall quality and the similarity to human translations regarding compound production, we computed the overall similarity of the influenced models' outputs to human translations. Figure 5.8 displays the BLEU score on reference A depending on prefix length. The score was computed after stripping the prefix from both the output and the reference translations, i.e. we removed the beginnings of the sentences that were given to the model from the reference translation. We performed the statistic on our first Transformer. The BLEU score increased with shorter prefixes (up to a length of 3) but then decreased rapidly (with a prefix of length 5), which surprised us.

We suspected that a longer prefix on average means a shorter sequence to be scored and for shorter sub-sequences of the MT output, it may be harder to hit the exact words of the reference. To check the behaviour of BLEU on shorter sentences, we computed the score for the original model for sentences grouped by their length to see whether the shorter sentences are on average less similar to the reference. Figure 5.9 shows the BLEU score of the original output sentences that are grouped by their lengths. The sentences with a maximum length of five words got a score of almost zero. The score grew for sentences containing at most seven words (but more than five) but descended further for sentences consisting of eight words. The BLEU score stabilized for sentences longer than 20 words at values of about 18.

The figure confirmed that shorter sentences had on average lower BLEU scores in our test set. Considering the conflicting trends of the decrease of BLEU in Figure 5.8 and the fact that BLEUs for shorter sequences are lower, we cannot make any concluding judgement about overall sentence quality vs. compounds in sentences we affected by prefix hinting. We can only refer back to the general observation (Figure 5.2) that overall quality seems to correlate well with the production of compounds.

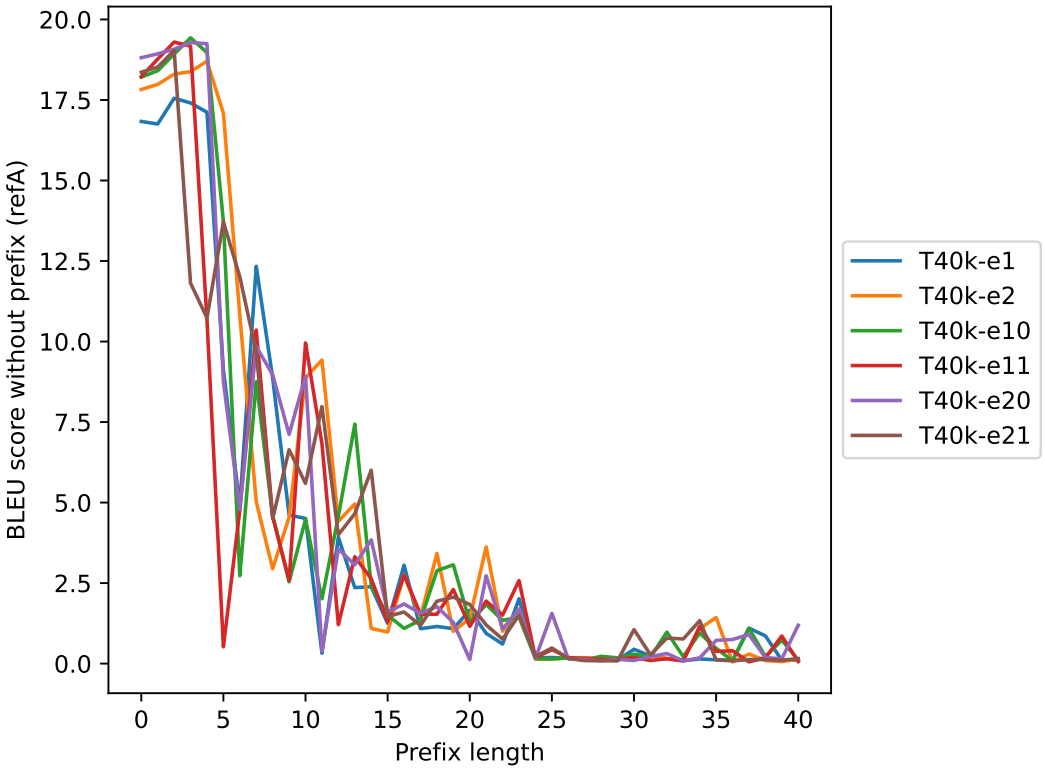


Figure 5.8: BLEU score for Transformer outputs with fixed prefixes

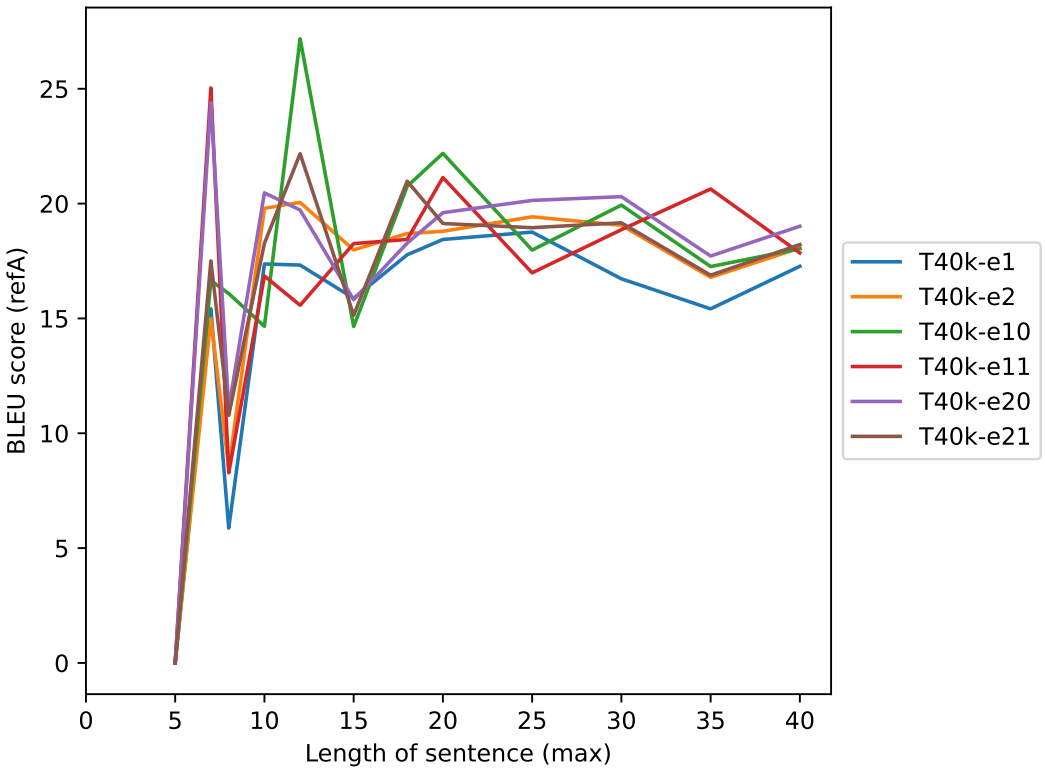


Figure 5.9: BLEU score for output sentences of our Transformer grouped by length

6. Conclusion

In the thesis, we performed various steps to explore the behaviour of Transformer models on German compounds in English-German translation. First, we studied the number of compounds in state-of-the-art systems submitted to the WMT21 conference. We focused mainly on constrained systems that were trained on fixed datasets. We discovered that human translators produce more compounds in their texts than any MT system in its output. The statistics were collected based on a fixed list of German nominal compounds extracted from GermaNet.

We also examined the words from the MT outputs that were not present in the training data because we found no newly generated German compounds based on the GermaNet list. We assumed that the MT systems operating on subword units are capable of generating novel words, including compounds. Our assumption was proven to be true based on our manual analysis of newly created words. Many of them were German compounds that were not included in any lexicon. Nevertheless, we found half of the novel nominal compounds in some articles on the Internet which confirms their validity. One third of the novel nouns were also present in reference translations.

As discussed, German has a very productive word formation system, and speakers often form new compounds based on the particular situation. We have shown that MT systems also produce new meaningful compounds that consist of existing constituents and are able to generate them in similar positions as humans do.

Even though the GermaNet list did not capture the productive word formation processes, it helped to group sentences where at least one system generated a compound. This made it possible to provide various analyses on compound production in various models. We trained our own Transformer model to explore the behaviour of the models in more detail.

Similar to SMT, the production of compounds can be influenced by word segmentation strategies. Therefore, we experimented with several setups of preprocessing, namely with the size and the number of BPE vocabularies. We found out that there was no big difference regarding the number of compounds when comparing setups with two separate dictionaries for each language and one dictionary for both languages. The model where we reduced the vocabulary size to one quarter generated much fewer compounds than the other setups based on the GermaNet list. We conducted a manual analysis of words that the system generated but were not in the training data and found several new compounds. Determining the novel compounds was conducted with the help of Google search. However, most of the novel words were not identified as proper German words. The hypothesis that forcing the model to split words into more subwords would lead to more compounds did not prove to be true.

We also investigated other factors that affect compound generation in the models. The frequency of the compound in the training data was shown to influence the probability of its appearance in the MT output. This dependency was even stronger for our vanilla Transformer than for the WMT21 systems. Although the BLEU score captures only the surface similarity of the outputs to the human reference, we examined whether there was a correlation between the

BLEU score and the number of found compounds in the outputs. We discovered that the correlation is rather high and better systems generate more compounds in their outputs.

We compared the results of our Transformer models with constrained systems from WMT21. It was not surprising that the WMT21 systems performed better in both the BLEU score and the number of produced compounds. The systems submitted to the shared task utilize enhanced models and not only the vanilla Transformer architecture as we do. However, none of the MT systems reached as high a number of generated compounds as we found in the human translations. It still remains an open question why people produce more compounds than MT systems. We proposed various possible explanations for it.

As described by Holtzman et al. [2019], outputs of systems that perform maximization-based decoding, such as beam search, are often *too* probable. We assumed that compound production could be influenced by language economy, which people tend to prioritize. The economy can be described in two ways: (1) people try to express their thoughts and needs in as few words as possible, and (2) they need to manage their language production load. It is easier to merge several parts of one thought into one compound than to express it with a multi-word expression that could also include prepositions, articles, and other words. Using prepositions is demanding because we need to choose them properly to express the exact thought, whereas the composition of the constituents is more vague and does not require exactness.

We proposed a modification to our model to deal with the point (2) economy. Assuming that the path generating a complex word is straightforward after the first subword, we computed the perplexity of the three following nodes in the search space and included it in the final score of the node. In other words, the model would consider the cost of future decisions already in the current choice. We only examined this idea on detailed outputs of a finished decoding process and did not implement it into the source code. We believe that adding this look-ahead to the model can make its behaviour more human-like and lead to the generation of more compounds. We would like to investigate this further in our future work.

To further explore the behaviour of our model on German compounds, we influenced it by forcing it to start with the first few words of the reference, and then we studied the outputs. We found that shifting the decoding algorithm to another subspace of the search space did not generally improve the generation of compounds. There was a small increase in the number of compounds for smaller prefixes. Although hinting the model with the first word helped the model to produce a better output in many cases, we found that it was more due to a search error than a systematic improvement. These results indicate that current models are inadequate with respect to compounds – less natural sentences with fewer compounds are scored higher.

In conclusion, we discovered that human translators produce more compounds than MT systems, and it is not easy to influence this tendency. The overall translation quality can indicate how well the system is capable of generating compounds in its outputs. Collecting German compounds in the translation based on a fixed list is not sufficient for any analysis because there are too many novel words created. We have shown that word segmentation influences the generation

of complex words. In our future work, we aim to build on our findings and modify the word segmentation strategies and MT models to bring their outputs closer to human translations.

Bibliography

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online, November 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.wmt-1.1`.

Irmhild Barz. German. In Peter O. Müller, Ingeborg Ohnheiser, Susan Olsen, and Franz Rainer, editors, *Word-Formation. An International Handbook of the Languages of Europe*, volume 4, pages 2387–2410. Mouton de Gruyter, Berlin, 2016.

Vladimír Benko. Aranea: Yet another family of (comparable) web corpora. In Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, editors, *Text, Speech and Dialogue*, pages 247–256, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10816-2.

Fabienne Cap, Alexander Fraser, Marion Weller, and Aoife Cahill. How to produce unseen teddy bears: Improved morphological processing of compounds in SMT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 579–587, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. doi: 10.3115/v1/E14-1061. URL `https://aclanthology.org/E14-1061`.

Joachim Daiber, Lautaro Quiroz, Roger Wechsler, and Stella Frank. Splitting compounds by semantic analogy. In *Proceedings of the 1st Deep Machine Translation Workshop*, pages 20–28, Praha, Czechia, 2015. ÚFAL MFF UK. URL `https://aclanthology.org/W15-5703`.

Philip Gage. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38, 1994.

Verena Henrich and Erhard Hinrichs. Determining immediate constituents of compounds in GermaNet. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 420–426, Hissar, Bulgaria, September 2011. Association for Computational Linguistics. URL `https://aclanthology.org/R11-1058`.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *CoRR*, abs/1904.09751, 2019. URL `http://arxiv.org/abs/1904.09751`.

Matthias Huck, Simon Riess, and Alexander Fraser. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 56–67, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4706. URL https://aclanthology.org/W17-4706.

Wolfgang Klein and Alexander Geyken. Das digitale wörterbuch der deutschen sprache (dwds). In *Lexicographica: International annual for lexicography*, pages 79–96. De Gruyter, 2010.

Philipp Koehn and Kevin Knight. Empirical methods for compound splitting. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary, April 2003. Association for Computational Linguistics. URL https://aclanthology.org/E03-1076.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/P07-2045.

C. Kunze and L. Lemnitzer. *Computerlexikographie: Eine Einführung*. Narr Francke Attempto Verlag, 2007. ISBN 9783823373155. URL https://books.google.de/books?id=gQT8DwAAQBAJ.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049.

Maja Popović, Daniel Stein, and Hermann Ney. Statistical machine translation of german compound words. In Tapio Salakoski, Filip Ginter, Sampo Pyysalo, and Tapio Pahikkala, editors, *Advances in Natural Language Processing*, pages 616–624, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-37336-0.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191,

Belgium, Brussels, October 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. SMOR: A German computational morphology covering derivation, composition and inflection. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/468.pdf.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

Milan Straka. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-2020. URL https://www.aclweb.org/anthology/K18-2020.

Sara Stymne. A comparison of merging strategies for translation of German compounds. In *Proceedings of the Student Research Workshop at EACL 2009*, pages 61–69, Athens, Greece, April 2009. Association for Computational Linguistics. URL https://aclanthology.org/E09-3008.

Sara Stymne and Nicola Cancedda. Productive generation of compound words in statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 250–260, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL https://aclanthology.org/W11-2129.

Kyoko Sugisaki and Don Tuggener. German compound splitting using the compound productivity of morphemes. In Adrien Barbaresi, Hanno Biber, Friedrich Neubarth, and Rainer Osswald, editors, *14th Conference on Natural Language Processing - KONVENS 2018*, pages 141–147. Austrian Academy of Sciences Press, 2018. doi: 10.21256/zhaw-4974. URL https://digitalcollection.zhaw.ch/handle/11475/14372. 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria, 19-21 September 2018.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Aleš Tamchyna, Marion Weller-Di Marco, and Alexander Fraser. Modeling target-side inflection in neural machine translation. In *Proceedings of*

the Second Conference on Machine Translation, pages 32–42, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4704. URL https://aclanthology.org/W17-4704.

Svetlana Tchistiakova, Jesujoba Alabi, Koel Dutta Chowdhury, Sourav Dutta, and Dana Ruiter. EdinSaar@WMT21: North-Germanic low-resource multilingual NMT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 368–375, Online, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.wmt-1.44.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

Marion Weller-Di Marco. Simple compound splitting for German. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 161–166, Valencia, Spain, April 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1722. URL https://aclanthology.org/W17-1722.

Marion Weller-Di Marco and Alexander Fraser. Modeling word formation in English–German neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4227–4232, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.389. URL https://aclanthology.org/2020.acl-main.389.

List of Figures

2.1 Composition of English test-set (number of articles) 12
2.2 Histogram of lemmas present in references according to frequencies in training data 13
2.3 Systems participating on WMT21 for English-German translation (C marks constrained systems and UC unconstrained) 14

3.1 Sets of German compounds appearing in outputs of two systems and a list of compounds from GermaNet 17
3.2 Frequencies of compounds in the training data and their appearance in constrained WMT21 systems' outputs 22

4.1 BLEU score for reference A during training of all four versions of the Transformer model computed by FAIRSEQ 29
4.2 BLEU scores during training of all four versions of the Transformer model computed by SacreBLEU 29

5.1 Counts of compounds in the system's output during the training of our Transformer compared to references and WMT21 systems 31
5.2 Comparison of overall translation quality to the number of produced compounds for constrained systems an our systems 32
5.3 Counts of words from our Transformer outputs that were not included in the training data 33
5.4 Frequencies of compounds in the training data and their appearance in outputs of first modification of our Transformer for all epochs 35
5.5 Counts of compounds in our Transformer depending on prefix length passed to the system 37
5.6 Count compounds in ref A depending on prefix lenght 38
5.7 Percentage of compounds from ref A comprised in Transformer outputs with fixed prefixes 39
5.8 BLEU score for Transformer outputs with fixed prefixes 42
5.9 BLEU score for output sentences of our Transformer grouped by length 42

List of Tables

2.1 WMT21 Training corpora - number of sentences and tokens . . . 12

3.1 Compounds appearance in English-German translations in WMT 21 (counts of all appearances of compounds and counts of sentences with compounds plus its subsets approved by reference translations) 19
3.2 BLEU scores of WMT21 systems – for reference A and all references excluding B, sorted descending by the score for refA 20
3.3 Statistics of words not seen in the training data for constrained systems 23
3.4 Statistics of words not contained in the training data for reference translations 25
3.5 Categories of unseen words produced by constrained systems according to manual analysis 26

4.1 Training setups of our Transformer model 27

5.1 Agregated statistic on changes of appearance of compounds in the output based on prefix 40

List of Abbreviations

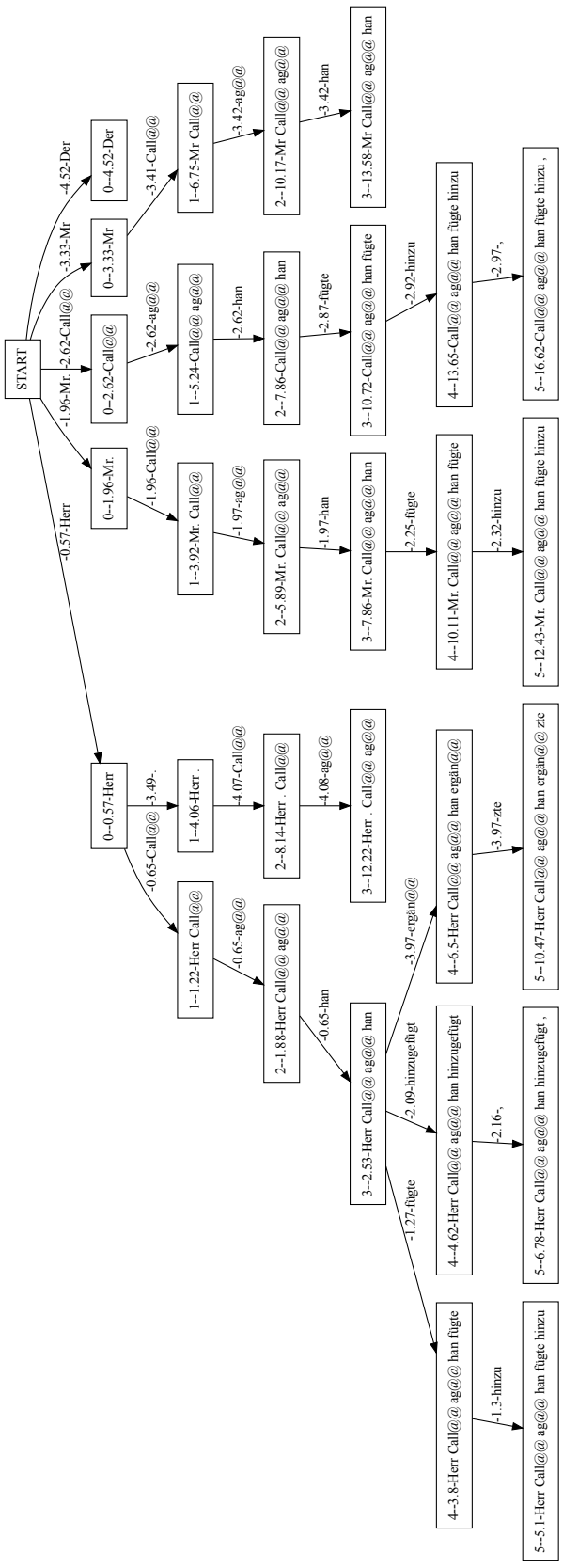
- NLP – Natural Language Processing
- MT – Machine Translation
- SMT – Statistical Machine Translation
- NMT – Neural Machine Translation
- WMT – Conference on Machine Translation
- POS – Part of Speech
- BPE – Byte Pair Encoding
- C – Constrained Systems
- UC – Unconstrained Systems

A. Attachments

A.1 Comparison of Lemmatization Methods

	UDPIPE		SPACY	
system	**# compounds**	**in refs**	**# compounds**	**in refs**
UC-ref-C	955		878	
UC-ref-D	946		882	
UC-ref-A	901		822	
UC-ref-B	878		815	
C-Nemo	842	735	774	664
C-UF	802	710	741	642
UC-metricsystem2	801		742	
UC-Online-B	798		736	
UC-Facebook-AI	796		755	
C-eTranslation	794	696	731	627
UC-VolcTrans-GLAT	792		733	
UC-Online-W	791		724	
UC-metricsystem1	790		732	
UC-metricsystem3	787		737	
C-WeChat-AI	783	707	718	632
UC-metricsystem5	783		724	
UC-VolcTrans-AT	782		712	
UC-Online-Y	776		713	
UC-happypoet	770		698	
UC-metricsystem4	769		714	
C-Manifold	768	666	710	601
UC-Online-A	767		704	
C-nuclear_trans	762	656	691	581
C-HuaweiTSC	761	673	705	609
C-UEdin	758	666	689	598
UC-Online-G	754		694	
C-P3AI	740	655	687	600
C-BUPT_rush	731	627	670	559
C-MyTransformer-e16	715	572	645	502
C-ICL	704	595	666	543
C-MyTransformer-e7	691	544	645	499
C-MyTransformer-e20	690	562	638	502
C-MyTransformer-e11	689	549	636	492
C-MyTransformer-e13	689	540	632	478
C-MyTransformer-e12	687	544	644	495
C-MyTransformer-e17	687	552	611	484
C-MyTransformer-e21	686	549	636	490

A.2 Beam Search Example (Original)



A.3 Beam Search Example (Forced Decoding)