Predicting German Compound Words using a Recurrent Neural Network

Edward Joseph Callow

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science in Computer Science and Systems

University of Washington

2019

Reading Committee:

Anderson Nascimento, Chair

Raghavi Sakpal

Martine De Cock

Program Authorized to Offer Degree:

Computer Science and Systems

University of Washington

Abstract

Predicting German Compound Words using a Recurrent Neural Network

Edward Callow

Accurate classification, morphological analysis and translation of compound words is a problem that has not been satisfactorily solved in many of its aspects. For example, as of the date of this paper, Google translates "Trittbrettunsterblichkeit*'*, a GCW meaning, in the English idom, the act of "riding on someone's coattails to achieve immortality" as "footboard immortality." This is a literal translation that does not capture the meaning. Conversely, when one tries to describe this idiom in an effort to get "Trittbrettunsterblichkeit", there is no way to get this word unless one inputs "footboard immortality", which makes no sense in English. Inputting "immortality achieved by riding on someone's coattails", which is a fairly accurate definition of "Trittbrettunterblichkeit" translates as the awkward phrase: "Unsterblichkeit, die durch das Reiten auf den Fellschwänzen eines Menschen erreicht wird." Clearly, constructing a GCW to match a concept in English, even when the word exists as a succinct native German word, is a problem. The goal of this thesis is to explore generation of GCWs, existing or non-existing, based on inputs of component root words. Although the methods explored may be adaptable to generation of

various words in various languages, the focus here is German compound words (GCWs), known also called Komposita. In particular, this thesis discusses the problem of predicting the correct linking element of the GCW.

To accomplish this a recurrent neural network (hereinafter ‘GCW RNN’) with Attention is used, trained upon the characters of the constituent words of the GCW in the training set. From this, a prediction is made as to the linking element. This report contains a description of the problem, the dataset, the model, and the results.

TABLE OF CONTENTS

Chapter 1. Introduction 7

Chapter 2. Literature Review 11

2.1 Rules of german compound words formation 11

2.1.1 Fugenelement in German compound words 13

2.2 Challenges of German Compound words 14

2.2.1 OOV Problem 14

2.2.2 Semantic Challenges 15

2.2.3 Morphological Challenges 17

Chapter 3. GCW-RNN Model 19

3.1 Project Objective 19

3.2 Model 20

3.2.1 RNN Architecture 21

3.2.2 Attention Model 23

Chapter 4. Experimental and Results 25

4.1 Settings 25

4.2 Dataset 25

4.3 GCW RNN Training Set 27

4.3.1 Implementation Details 27

4.4 Results 29

Chapter 5. Conclusion & Future Work 33

5.1 Observations based on Preliminary Work 33

5.2 Next Steps 33

Bibliography 34

Chapter 1. INTRODUCTION

Accurate classification, morphological analysis, translation and generation of compound words is a problem that has not been satisfactorily solved in many of its aspects. Many languages contain compound words, so optimal translation requires that occasionally words in one language be mapped onto a compound in the target language, especially where a succinct and accurate compound exists, rather than to tranlsate it into an accurate but clumsy phrase.

For example, as of the date of this paper, Google translates "Trittbrettunsterblichkeit*'*, a German Compound Words (GCW) meaning, in the English idom, " immortality achieved by riding on someone's coattails" as "footboard immortality." This is a literal translation that does not capture the meaning of the GCW in the English. Conversely, when one tries to describe this idiom in an effort to get "Trittbrettunsterblichkeit", there is no way to get this word unless one inputs "footboard immortality", which makes no sense in English. Inputting "immortality achieved by riding on someone's coattails", which is a fairly accurate definition of "Trittbrettunterblichkeit" translates as the awkward phrase: "Unsterblichkeit, die durch das Reiten auf den Fellschwänzen eines Menschen erreicht wird." Clearly, constructing a GCW to match a concept in English, even when the word exists as a succinct native German word, is a problem.

This thesis explores classification and prediction of the morphology of existing GCWs using neural models. The goal is to provide both linguistic insight into the formation of GCWs and also a starting point towards a different, and more effective approach of translating any source language to GCWs. For example, it may eventually be possible to translate an English phrase such as " immortality achieved by riding on someone's coattails " into a single German compound such as "Trittbrettunsterblichkeit", which would both save excess verbiage and create a more native German look-and-feel.

Also, this project acts as a foundation that can be adaptable to one of the many other languages with an abundance of compound words. Also, using the approach suggested here, which looks at word generation in the abstract, it would also be possible to create new compound words that do not yet exist. As a starting point for this more general problem, creation of compound words in the German language is examined, due to the large numbers of compounds in German and the problems this has created for other aspects of machine translation.

The structure of a GCW is most commonly simply **A** + **B** (where 'A' and 'B' denote the words in the compound). However, there is often a letter or group of letters between the words, which may or may not add semantic information, known as linking elements, resulting in the form **A + L + B** (where 'L' denotes the linking element). So, for example, in the word "Bilderrahmen" ('Bild'(picture)+ 'Rahmen'(frame) = picture frame),the "er" between the 2 words is the linking element, and is superfluous from a semantic perspective [3]. Below is a table of GCWs demonstrating the component words and the linking elements for a few select 2-word compounds:

Table 1: Examples of German Compound Words Split by Linking Elements

GCW	**First Word**	**Link**	**Second Word**
Bilderrahmen (picture frame)	Bild	er	Rahmen
Schneeweiss (snow white)	Schnee	None	Weiss
Arbeitstier (workaholic)	Arbeit	s	Tier
Hundemüde (dog tired)	Hund	e	Müde
Landesgeld currency	Land	es	Geld
Gedankenfreiheit (freedom of thought)	Gedank	en	Freiheit

As the examples in Table 1 demonstrate, the linking element between words in a compound may or may not exist, and if they exist, several are possible. Therefore, in order to accurately predict the form of a GCW, it is crucial to predict the correct linking element. Sometimes, the linking element has some grammatical significance; for example, 's' between words might indicate possessiveness, which is expressed by the genitive case; however, this is not always true, and often there is no link, even if there is clearly a possessive relationship between the two words [3]. For example, "Landesgeld" in the table above could be described as the **Geld** (money/gold) belonging to the **Land** (country). However, "Jugendsunde" (youthful folly), could also be so described – as the **Sunde** (sin) belonging to **Jugend** (youth). In this case, the second noun is feminine, and the genitive declension for feminine nouns in German does not change the form. On the other hand, in plenty of cases, such as in the case of "Schreibtischcomputer" (desktop computer) the first noun **Schreibtisch** (desk) is masculine and the possessive would be formed by adding an 's', but none is added to form the compound word, even though there is arguably a possessive relationship between the words. Therefore, in order to accurately create a German word, and apply it to MT, the model must be able to make these subtle grammatical analyses and be able to classify exceptions to general rules that have identifiable patterns.

The purpose of this thesis is to outline the process of how a GCW might be assembled not from the perspective of translation from a particular language into German but from the perspective of word creation as a general case. Due to the success of RNNs with MT and the sequential nature of compound words, we use an RNN to form compounds and predict the linking elements [4]. In this paper we describe the implementation our model with respect to correct word formation where the constituent words are already known but the link (see Table 1 above) must be predicted. We

also compare our model to a Naïve Bayes model using character n-grams and present the accuracy of each model. We also discuss possible future expansion of this current project.

Chapter 2. LITERATURE REVIEW

Compound words exist in many languages and account for a significant portion of the vocabulary in most Indo-European languages, as well as Chinese. Compounding, as well as borrowing from other languages, are among the most prolific tools a language has for creation of new words. However, the focus of this thesis is GCWs in particular.

2.1 RULES OF GERMAN COMPOUND WORDS FORMATION

GCWs can be classified into different types, depending on the semantic relationship between the words, described in Fleischer 1995 at 125-132 [3]. These types include: (A) subordinating compounds, (B) coordinating compounds, (C), exocentric compounds, and (D), copulative compounds [5], as further described below:

Table 2.1 Classification of Types of Noun Compounds

Compound Type	Definition	Example
Subordinating	'a kind of X'. The preceding words (aka 'determiners') refer to the kind of primary word.	'Haustür [house door].' A 'house key' is a subset of key. In English 'smalltalk'.
Coordinating	Both parts have equal weight. The words have different descriptions for the same object.	"nasskalt" - meaning dank, 'wet' + 'cold'. Here, the meaning is somewhere in the intersection of the two words.in English 'maidservant'.
Exocentric	Two words combine to form a new word of a different meaning and often different grammatical category.	'Taugenichts - ne'er-do-well, "ne'er" + "nothing". The meaning is outside the scope of either word, referring to a kind of person. In English "smart alec", "skinhead".
Copulative	The meaning is a "sum" of the two words. "Both parts	"Schleswig-Holstein." The copulative compound always

	are of equal rank designating separate entities which together produce a new entity" [6].	combines the same parts of speech. In English, "sleepwalk".

Of the above-referenced kinds, the subordinating compound is the most common and the easiest to form. For example the meaning of "hiking shoes" (a subordinating compound) is clear and easily understood to a non-native English speaker, but "White Collar" (an exocentric compound which refers to something that is neither white nor a collar) is not.

For the task of MT, it may be necessary to classify the semantic category of the compound, above, in order to apply the correct translation analysis. As Sorokin, et al 2017 at 58 describes, various efforts have been made to classify compounds based on semantic relationship with different approaches [7]. In general, these approaches do not classify according to the categories above but use ad-hoc methods tailored towards translation. For example, Sorokin proposes a model employing a support vector machine (SVM) to classify according to prepositions – so for example in the category for the preposition 'for', he places 'Autohaus' (house FOR cars, i.e. car dealership); an example for 'of' is Schneehaus (igloo – lit. house of snow) [7]. From a translation perspective this certainly makes sense, as classified compounds would lend themselves to paraphrasing or approximate translations in the case of an out of vocabulary (OOV) word (see Section 3.2.1 below). With this approach, Sorokin achieves approximately 60% accuracy, from a baseline of 22.66%. However, this preposition-based approach most likely cannot account for exocentric compounds, whose meaning is not fully captured by the words themselves (and by extension their interrelationships). Therefore, while this approach may be a good starting point, it would be preferable to find an approach that can account for all kinds of GCWs.

2.1.1 *Fugenelement in German compound words*

Based on our own analysis of the dataset (described in Section 4.2), approximately 30% of all German compound words include a linking element, known in German as a "Fugenelement" (FE or 'link'), which is a bound morpheme that connects the individual stems with each other. In German, these are *-e-*, *-s-*, *-es-*, *-n-*, *-en-*, *-er-*, and *-ens-*. When no FE is visible within the word, linguists assume the existence of a null morpheme Ø – an invisible suffix.

The majority of GCWs are noun compounds. However, other compounds, such as adjective-noun (e.g. "superman"), verb-noun (e.g. "thinktank"), and adjective-verb (e.g. "to supersize"), are also possible. The following table outlines some general cases for the linking element in noun compounds. Fleischer 1995 describes these morphological tendencies [3]:

Table 2.1.1 General Cases for Linking Elements in Nouns

Link	**Situation used**	**Example**
e	When the plural of the first noun adds an -e-	Die Hundehütte (der Hund -> die Hunde)
er	When the first noun is either masculine or neutral and is pluralized with -er-	Der Kindergarten (das Kind ->die Kinder)[1]
n, en	When the first noun is feminine and is pluralized -en-	Der Birnenbaum / the pear tree (die Birne -> die Birnen)
s	When the first noun ends in either *-heit, keit, -ung*	Die Gesundheitswerbung / the health advertising
s, es	For some nouns that end in -s- in the genitive case.[2]	Das Säuglingsgeschrei / the newborn's cry (des Säuglings)

[1] However, consider Wörterbuch ('dictionary'), for which "Wortbuch" is also an acceptable compound, perhaps reflecting the fact that "Wort" has 2 acceptable plural forms, "Wörter" and "Worten".
[2] However, the ending 's' and 'es' has also cases where it shouldn't according to this logic. For example, 'Liebesbrief' (love letter). "Liebe" is feminine, so its genitive is not formed with 's'. Exceptional uses for each of these rules exist in the language.

The table above illustrates some general tendencies for noun compounds. For verbs and other parts-of-speech there are fewer rules; however, there is an exception for verbs:

Table 2.1.2 Linking Elements in Verb + Noun Compounds

Link	Situation used	Example
e	After many verbs with a stem ending in b, d, g, or t	Der Liegestuhl / the lounge chair

While these compounds are normally written as a single, continuous word, hyphenated '-' forms occasionally appear in new coinages of German compound words ("Der Cyber-Diebstahl" / cyber-theft). Hyphens are also used when a series of compounds share a "primary word" ('Halsschmerzen' [Sore Throat] 'Kopfschmerzen' [Headache] and 'Gliederschmerzen' [Body ache] are compressed to 'Hals-, Kopf- und Gliederschmerzen'). Hyphenated forms are relatively recent, and do not follow perceivable patterns, which poses a challenge for word formation, as further discussed in Section 3.2, below.

2.2 CHALLENGES OF GERMAN COMPOUND WORDS

2.2.1 *OOV Problem*

As GCWs are often created on an ad-hoc basis by speakers or writers to describe a new phenomenon (e.g. "time-traveler"), many may not be listed in the dictionary. In computational linguistics, this is known as the out-of-vocabulary problem (OOV), i.e. the phenomenon that the word sought to be translated is not present in the dictionary or corpus. As Goldsmith et al. 1998 notes, GCWs have a high rate of OOV, which creates a significant problem for translation into German [8]. This also means that for a truly authentic rendition of German, the ability to invent

appropriate GCWs is important. So far, at best translators find matches for phrases that match existing GCWs.

In the context of speech recognition, Geutner 1995, also discusses the importance of separately identifying and translating GCWs [9]. He notes that specific treatment of compounds decreases a substantial part of the OOV problem, and that compounds are a primary cause for OOV being more significant in German than in English. Berton et al. 1996 also describe work aimed at improving OOV responses of a speech recognition system by allowing the language-model to include compounds [10]. This previous work demonstrates that the OOV problem requires special treatment of compound words in the case of MT. In particular, it shows that a dictionary-based approach is not sufficient to solve this problem.

Likewise, the approach taken in this thesis is to treat GCWs without reference to a dictionary and to look at specific problems which must be solved in order to create authentic GCWs as the situation may demand. The primary application of this approach will be translation of English (or other languages) into German; however, it may also be useful for translaton from German. For example, generating a GCW provide a useful check for such translations by reverse-engineering them. If the translator translates "trittbrettunsterblichkeit" as "footboard immortality", subsequently inputting "footboard immortality" and trying to create the corresponding GCW may show that the results do not match and therefore the probability of a correct translation is lower.

2.2.2 *Semantic Challenges*

In the case of GCWs, particularly exocentric compounds (discussed in Section 3.1 above), there may exist complex meanings which may not have a clear relationship with the meanings of

the words themselves. Consider English exocentric compounds such as ‘white collar’ or ‘have-nots’. These words pose challenges for MT as well.

For example consider the exocentric compound *‘Trittbrettunsterblichkeit’*: This coinage is made up of six separate elements: "der Tritt" (*step*) and "das Brett" (*board*) combine to fashion "das Trittbrett" (*running-board*, the foot-board that runs along the side of certain trucks and antique automobiles). German usage has given this word a metaphorical connotation: "hanging on" or "freeloading," especially in the term of "Trittbrettfahrer" (literally: *running board rider*). To "Trittbrett" the author adds "Unsterblichkeit" (*immortality*). That term is derived from "sterben" (*to die*); by adding the suffix "-lich" to the stem "sterb-", we get the adjective "sterblich" (*mortal*). To that comes the suffix "-keit", forming the noun "Sterblichkeit" (*mortality*). The prefix "un-", just as in English, produces the opposite meaning (*immortality*). Finally, the author combines "Trittbrett" and "Unsterblichkeit" to create a spontaneous coinage that will appear in no dictionary but is comprehensible to native German speakers: "Trittbrettunsterblichkeit" is *an immortality that is achieved by riding in someone's wake (or on someone's coattails)*. The English translation necessarily lacks both the compactness - and the humor - of the German original. From a translation perspective, moreover, one would be hard-pressed to derive the correct translation by simply employing a dictionary, without intimate knowledge of German and the cultural context that led to the minting of the word. However, there is no adequate way to derive "Trittbrettunsterblichkeit" through standard translation tools. For example, if one inputs “an immortality that is achieved by riding in someone’s wake,” the output is a word-for-word translation rather than the single compound which expresses this concept. Strangely, if one inputs “footboard immortality” the word is derived, although “footboard immortality” is nonsense in

English." Most compound words are not this complex, but as the example illustrates, there are numerous potential subtleties required for formation and translation of compound words.

Sorokin, et al 2017, discussed in Section 3.1, above, uses a semantic classification of compounds that may be a starting point to the solution of this problem [7]. Sorokin uses an SVM architecture to classify words in the correct semantic category based on prepositions (e.g. 'for', 'in', 'of'), in order to evaluate which preposition best describes the relationship between the words in the compound [7]. Cap 2014 similarly applies a semantic filter [11]. However, these models are not sufficiently complex to account for the metaphorical aspects of certain exocentric compounds, and this will have to be addressed in order to successfully classify and create words such as "trittbrettunsterblichkeit." In the case of formation of a GCW, the challenge is to select words with not just literal meanings but in some cases clever allusions or figurative language, such that, when appropriate, the allusion evokes the intended meaning.

A solution of this problem is not attempted here, but in the case of word creation, ultimately, classification and prediction of forms based on linguistic categories such as exocentrism would be necessary if, for example, one wanted to create a word with a description more sophisticated than (e.g. "house for dogs" → "doghouse").

2.2.3 *Morphological Challenges*

Goldsmith et al. 1998 described work aimed at developing a morphological analyzer to determine the morphological property of individual German noun stems. The results supported the strategy of using large-scale natural language corpora as a source for automatic processing and as a means to gather specific lexical information. The work only supported noun-noun compounds and not other compound word combinations [8].

If one attempts to create new GCWs, it is necessary both to identify the component words and to choose the correct linking element. In the context of machine translation, this challenge is confronted when one attempts to translate another language to German, for example English to German. See Cap 2014 at 171-186 [11].

Previous studies of translation to German compounds which have been examined have not analyzed the treatment of the Linking element in depth. For example, the technical process of creating links is not specifically described in Cap 2014 [11] and the accuracy of results for this process is compounded into the overall accuracy of translation in terms of BLEU scores. Presumably this is because ultimately the translation does not generate new words and can check the result against a dictionary for spelling inconsistencies with a traditional edit-distance analysis. Creation of new words without reference to a dictionary requires that the correctness of linking elements be considered independently.

If we treat creation of compound words as a word generation task from an NLP perspective, there are numerous examples of the use of neural networks to generate text in general. A famous example is Vinyals, et al. 2014 [12], which uses neural networks to generate a caption appropriate to an inputted picture. Also, in pop culture, there are pop culture examples, such as Deep Drumpf, a neural network used to generate tweets in the style of president Donald Trump.[3] In a sense, this project is similarly about word generation, except that it occurs at the morphological level and is concerned with generating words from basic morphemes rather than sentences. Stated differently, the goal is to generate a single GCW from an input of multiple words, possibly a phrase, rather than a sentence or group of words of equivalent length.

[3]Available at https://twitter.com/deepdrumpf?lang=en. Last checked on 7/19/2019.

Chapter 3. GCW-RNN MODEL

3.1 PROJECT OBJECTIVE

The objective is to implement a deep learning model which can successfully form GCWs, in particular by predicting the correct link. As previously discussed, formation of GCWs poses various challenges, including: 1) morphological, i.e. determining the correct linking element between the words of the compound; and 2) semantic (identifying the correct words and relationships between them).

Our research objective is to address the morphological problem of generating the linking element with a deep learning architecture.

In this report, we provide details regarding the design and implementation details of a GCW-RNN which addresses the morphological problem. Table 3.1 shows the input and output for this model.

Table 3.1: Sample Input/Output[4]

Input	**Output**
[Bild, Rahmen] (picture, frame)	Bild**er**rahmen (picture frame)
[Absicht, los] (intention, less/without)	Absicht**s**los (unintentional)
[Gedank, Freiheit] (thought, freedom)	Gedank**en**freiheit (freedom of thought)
[Adsorption,kalori, metrie] (adsorption calorie, metry)	Adsorption**s**kalorimetrie (adsorption calorimetry)

[4] Note that the English is not actually part of the input/output but is provided for convenience.

3.2 MODEL

Classifying the correct linking element of a GCW can be seen as a sequential task, wherein the sequence predicted is a sequence of characters comprising the entire GCW, inclusive of its linking element(s), if any. By classifying the task as sequential, it is implied that the probability of any character occurring is dependent upon the other characters in the word. For example, in general it is more likely for a consonant to appear after a vowel in English, and vice-versa. We chose to go with RNN (including LSTM), as RNN is an effective model for sequential tasks with a sufficiently large dataset, as demonstrated by Sepp Hochreiter and Jurgen Schmidhuber in their 1997 work [14]. Andrej Karpathy's influential blog describes implementation of a character-level RNN [15].[5]

We also chose to use the Attention model, which is a variant of the RNN framework. In particular we used a bidirectional RNN (BRNN) with Attention, which is commonly used for sequential analysis, particularly machine translation, since its introduction by Bahednau et. al in 2014 [16], as well as the standard unidirectional, which was also implemented with Attention. We refer to both interchangeably as a "GCW-RNN."

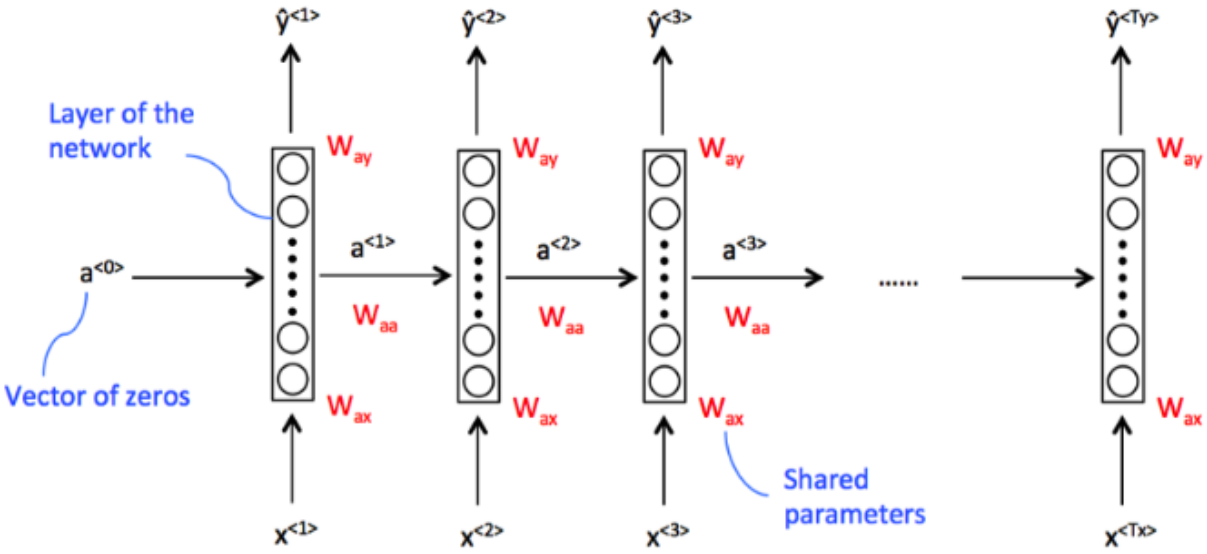
Both bidirectionality and the Attention model are variants of the standard RNN and include the same parameters. Therefore, we first describe the RNN and then proceed to describe bidirectionality and Attention.

[5] See also Karpathy's character RNN implementation, available at https://gist.github.com/karpathy/d4dee566867f8291f086.

3.2.1 *RNN Architecture*

The architecture for the standard RNN is depicted in Figure 3.3.1 below:

Figure 3.2.1 Standard RNN architecture[6]



A given time step *t* is calculated as follows:[7]

$a^{<t>} = g(Waa * a^{<t-1>} + Wax * x^{<t>} + ba^{8})$ (1)

$\hat{y}^{<t>} = g(Way * a^{<t>} + by)$ (2)

- $X^{<t>}$ represents input into the network at time step *t*, represented by one-hot vectors, times *M* training examples, producing a matrix of size 31 x M, where M represents the number of training examples and 31 are the possible character inputs. A one-

[6] This figure was taken from Michele Cavaioni's blog on RNNs, found at https://medium.com/machine-learning-bites/deeplearning-series-sequence-models-7855babeb586

[7] Appendix 1 contains the dimensions and values of the variables below for our model.

[8] 'ba' here is bias, which shifts the activation function left or right as necessary to fit the data

hot vector is created assigning a number to each possible letter (in German there are 29 letters). Additionally, characters for spaces (padding) <pad> and potentially unknown characters <unk> are added to accommodate errors and differences in lengths of input and output. For example, a one-hot vector representing the letter 'b' would be [0100..00], with twenty-five 0s and a 1 in the second position.

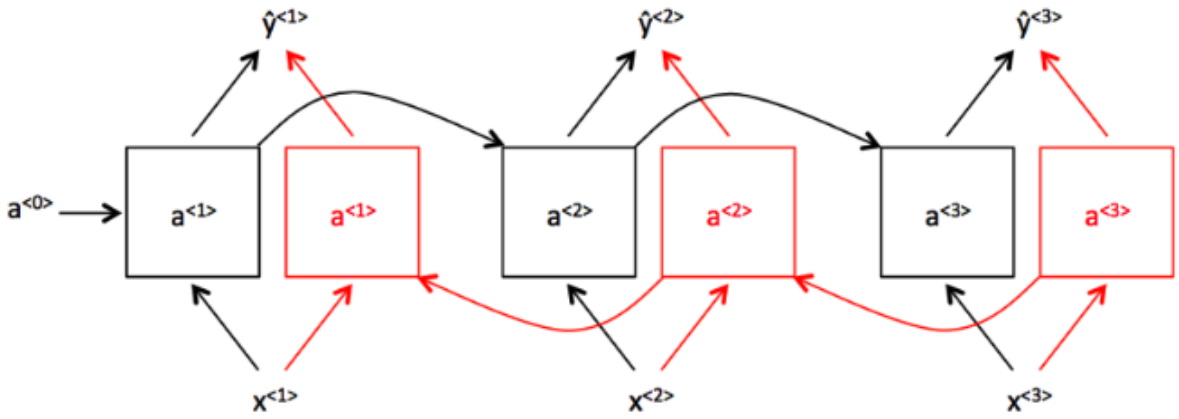
- The activation values $a^{<1>}$…$a^{<Tx>}$ (each layer here represents a single output character predicted in the word) are calculated by using a ReLU (here denoted by 'g') activation function on the parameters Wax and Way, which are constant throughout the layers.
- Waa is a weight *matrix* of dimensions A * A, where A is the specified output length for the entire model (in this case a word with no more than 25 characters).
- Wax is a matrix of dimensions A * X, where *X* is input (here one of the 31 possible German characters in a 1 * 31 one-hot vector).
- $Y^{<t>}$ is the predicted output for layer t, and the activation for this layer is softmax regression (also denoted by 'g' above).
- Gya is a matrix of dimensions Y * A, where y is a vector with possible outputs (here 31 possible German characters).

The key to the RNN is that $a^{<t-1>}$ is passed into the calculation of $a^{<t>}$. This allows the RNN to learn based on the previous parts of the sequence. This makes it is easier for a neural network to predict the letter is 'r', if it knows the preceding letters are 'daughte'.

As discussed in Section 3.2, we are using a bidirectional RNN ("BRNN") as well as the standard RNN. A BRNN is exactly the same as an RNN, except that it moves in both directions,

and therefore, the output of $\hat{y}^{<t>}$ will not only be predicted based on the preceding values in the sequence but also on the values that follow it (See figure 3.2.2 below). For example, suppose you were trying to predict the second letter in the word "itch". The probability of 't' being the correct letter is influenced by 'ch' being the following letters and 'i' being the preceding letter. Figure 3.3.2 illustrates the bidirectional flow of the BRNN.

Figure 3.2.2 Bidirectional flow of parameters in BRNN[9]



3.2.2 *Attention Model*

An additional problem for sequence models is weighting the various elements in the sequence. For example, if you are predicting the word "rice" in "John likes beans and rice, and eats this every Saturday before practicing the violin", the word "violin" is much less helpful than the word "beans" for predicting "rice" as the correct word in the sentence. A pure RNN or BRNN would weight all elements equally, and this leads to a decrease in accuracy or BLEU score for

[9] This figure was taken from Michele Cavaioni's blog on RNNs, found at https://medium.com/machine-learning-bites/deeplearning-series-sequence-models-7855babeb586

longer sequences, as elements closer to the value sought to be predicted generally have greater relevance. The Attention model solves this by taking a weighted average of each timestep $a^{<t>}$, giving greater weight to elements near time-step *t*, as described in Bahdenau, et al 2014 [16]. Hence, we use the Attention Model for both RNN and BRNN.

Chapter 4. EXPERIMENTAL SETUP AND RESULTS

4.1 SETTINGS

Both a standard RNN with Attention and a BRNN with Attention were tested for the prediction of linking elements. Using a BRNN improved the accuracy by approximately 0.5% overall. See Table 4.4.1 and 4.4.2, below, for a full listing of the accuracies for the various linking elements of this model.

Minimal pre-processing was required. Preprocessing consisted of making words lower case and then vectorizing all words into one-hot vectors. For back propagation, cross-entropy loss and Adam optimization were utilized, as these are generally the most effective parameters.

The corpus was split into training and test sets, and the process was repeated using a 10-fold cross validation for both models.

4.2 DATASET

The corpus used is a subset of the Tiger Corpus (version 1), maintained by the University of Stuttgart, which is an NLP corpus consisting of approximately 700,000 tokens (40,000 sentences).[10]

The XML version of the corpus was processed by selecting only words that were compound. To determine if a word was compound, Daniel Naber's Java German word splitting plugin was used,[11] which is a dictionary-based algorithm. It splits compound words into their constituent parts by detecting whether any subset of the word is in the corpus. For example, if "pineapple" were the word analyzed, the algorithm would split it into ("pine" and "apple"),

[10] available at https://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.en.html.
[11] Available on github on the repository danielnabler/jwordsplitter.

provided either “pine” or “apple” were in the dictionary. Only words where splits occurred were retained from the original corpus. The resulting corpus was thus a corpus of all compound words identified by the word splitter. Of these words, only unique values were kept, as words were duplicated in the original to represent different grammatical functions. The resulting data contains 27,018 unique compound words. The corpus includes not only Noun + Noun compounds but compounds of all parts of speech (POS) types.

From these remaining 27,018 words, hyphenated words were also removed, resulting in a corpus of 24,819 words. Hyphenated compounds were removed due to low accuracy (approximately equal to random guessing in the initial trials). Use of the hyphen is somewhat recent in German, and particularly prevalent in the 1990s, and based on initial experimentation, the use does not follow any predictable character pattern, unlike other linking elements.

Within the corpus, the data was not balanced among the linking elements. Approximately 68% of the GCWs contain no linking element. The next largest category is ‘s’ (24%), then ‘n’(5%), then ‘en’(2%). The remaining links comprise approximately 1%. Thus, 68% accuracy is used as a baseline metric in accordance with the Zero rule algorithm. Table 4.2 below shows the breakdown of linking elements in the training set.

Table 4.2 Training Set by Linking Element

Link	# of training examples
Null	15240
‘e’	66
‘en’	392
‘ens’	1
‘er’	88
‘es’	10
‘n’	1107
‘s’	5470

4.3 GCW RNN TRAINING SET

From the corpus, the model analyzed input data (X), containing each word in the compound, in order to predict the full GCW (Y), which includes the linking element, if any. For example, for “absichtslos” (Y), the X input data are GCWs are “[absicht, los]” (without the linking element ‘s’).

Using data for supervised learning has the inherent limitation that it can only verify the accuracy for the prediction of existing words. In order to fully test the model’s ability to generate GCWs, performing a user study with native speakers to test the efficacy of the model would be necessary.

4.3.1 *Implementation Details*

We used the Keras API for the Tensorflow library in order to implement our model and chose LSTM rather than GRU. The relevant code, including parameters, from the Keras implementation of our LSTM model is as follows:

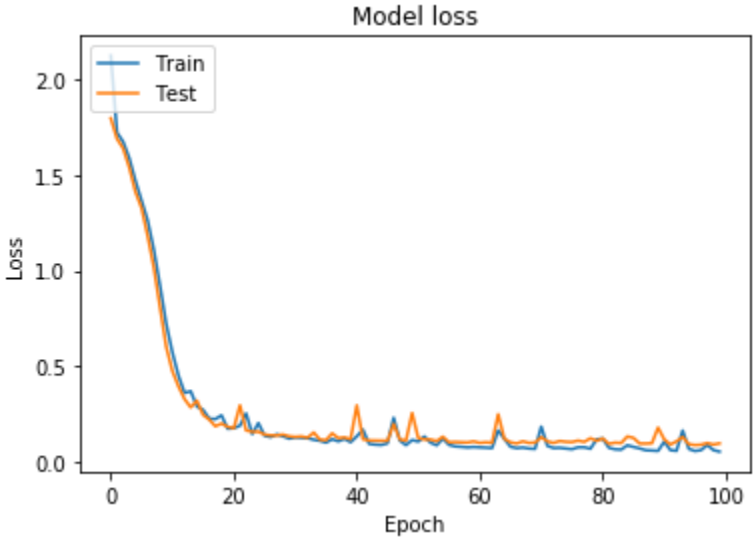
Figure 4.3.1: Keras Implementation

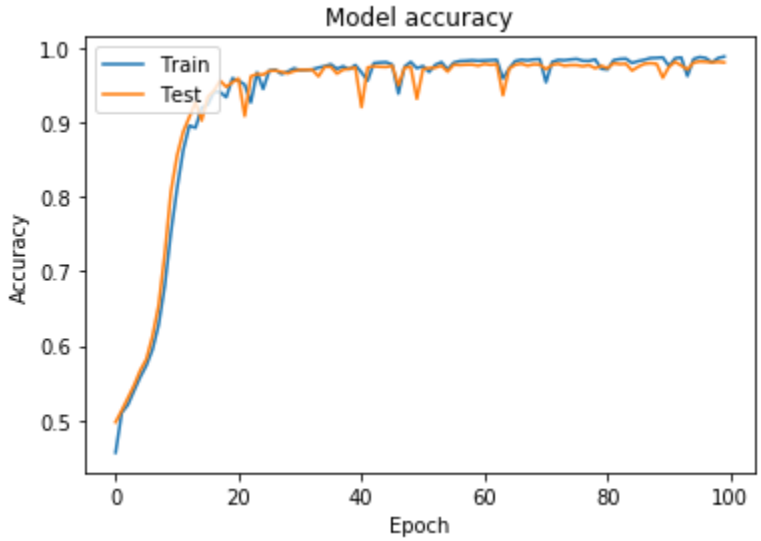
```
model = Sequential()
model.add(LSTM(128, input_shape=(Tx, 33),
return_sequences=True))
model.add(AttentionDecoder(128, 32))
model.compile(loss='categorical_crossentropy',
optimizer='adam', metrics=['acc'])
model.fit(Xoh, Yoh, epochs=100, batch_size=256)
```

LSTM was chosen over GRU based on a higher accuracy after testing both models. Because the model predicts individual characters and there are 31 options for each prediction, standard accuracy was chosen over metrics such as precision and recall, due to the difficulty in assessing false positives.

As the loss and accuracy curves demonstrate, approximately 100 epochs results in the highest accuracy for the model. The discrepancy in the accuracy result from the results reported below is because Keras calculates accuracy at the character level, so a partially correct word from its perspective may contain 50% accurate predictions, which for our purposes is completely incorrect.

Figure 4.3.2 Model Loss and Accuracy for Training and Testing





4.4 RESULTS

The model was quite accurate at predicting the correct linking element. This suggests that the linking element has strong morphological predictability based on the sequence of letters. Furthermore, as results did not significantly improve with a bidirectional network, it is likely that the linking element is not determined by the word following the link.

The dataset taken as a whole was randomly shuffled into a 90% training and 10% testing dataset, and this process was repeated 10 times using 10-fold cross validation. The tables below summarize the accuracy by linking element in the training and the test datasets for. For purposes of accuracy reporting, linking elements ‘ens’ and ‘es’ were discounted because there were an insufficient number of training samples (1 and 10, respectively) to create statistically significant results, though it is noted that on the face of it, the model appeared to be able to predict these elements as well. Also, for purposes of the model, hyphenated words were not considered as linking elements. Although German does have a number of words connected by hyphen, this is a fairly recent phenomenon and often used in the context of loan words. We did report the results

of our model on hyphens as linking elements, but is should be noted that these results were not above baseline accuracy.

As a sanity check, a Naïve Bayes classifier using bigrams and trigrams was also implemented. Unlike the sequential network, the Naïve Bayes classifier did not predict the entire word but merely from a choice of linking elements. The overall accuracy results are reported in the table below:

Table 4.4.0 Summary of Accuracy Results

Model	**Sequence**	**Performance (train)**	**Performance (test)**	**epochs**	**lr**	**batch**
Naïve Bayes	bigrams	81.0%	80.8%	-	-	-
	trigrams	84.6%	84.0%	-	-	-
RNN	LSTM with Attention	91.2%	88.3%	100	.001	256
	Bidirectional LSTM with Attention	91.6%	89.0%	100	.001	256

As Table 4.4.0 indicates, the Naïve Bayes model also performed better than baseline. Trigrams performed better than bigrams in the Naïve Bayes model. However, both RNN models significantly outperformed the Naïve Bayes model. Technically the results of the BRNN were slightly better than the other RNN model, but the difference was not great enough to be statistically significant. This may indicate linguistically that the linking element is mostly, if not exclusively, determined by the preceding word rather than the word(s) following it.

The following tables summarize the accuracy for the individual linking elements.

Table 4.4.1 Linking Element Prediction Training Accuracy by Link

Link	**Training accuracy GCW-RNN (Bidirectional)**	**Training accuracy GCW-RNN (Non-Bidirectional)**	**# of training examples**
	% Acc	% Acc	
Null	92.4	91.2	15240
‘e’	80.4	83.5	66
‘en’	96.3	95.4	392
‘er’	93.2	93.2	88
‘n’	98.6	97.6	1107
‘s’	88.7	92.3	5470
TOTAL	**91.6**	**91.2**	**22337**

Table 4.4.2 Linking Element Prediction Testing Accuracy by Link

Link	**Testing accuracy GCW-RNN (Bidirectional)**	**Testing accuracy GCW-RNN (Non-Bidirectional)**	**# of testing examples**
	% Acc	% Acc	
Null	88.1	87.4	1714
‘e’	60.4	60.4	10
‘en’	92.8	92.8	38
‘er’	86.4	86.4	14
‘n’	95.7	95.7	133
‘s’	89.2	89.2	610
TOTAL	**89.0**	**88.3**	**2519**

Due to the small data size, ‘e’ and ‘er’ are also of questionable significance. Nevertheless, the high accuracy suggests the model has some predictive power for those links as well as the more prevalent links such as ‘s’, ‘n’, ‘en’ and ‘Null’.

Based on the results in tables 4.4.1 and 4.4.2, it seems that it is possible to predict the correct linking element in German with a high degree of accuracy. As noted above, if the hyphen is treated as a linking element, accuracy is not above baseline for the hyphen. As

linguists do not all consider the hyphen to be a linking element, and because it does not appear necessary to use the hyphen to create new compounds, we removed the hyphenated words; however, in the future, this assessment may be revised, and it may be necessary to create a model which can predict hyphens.

Chapter 5. CONCLUSION & FUTURE WORK

5.1 OBSERVATIONS BASED ON PRELIMINARY WORK

The goal of this research was to generate GCWs and to identify what rules apply to the generation of them, if any. We applied the GCW-RNN model discussed in Section 3.2. Based on the results thus far, it is our conclusion that the GCW-RNN is able to generate linking morphemes (if one does not consider the hyphen as a linking morpheme) with a degree of accuracy considerably above baseline for GCWs of all kinds, irrespective of the length or the parts of speech in the constituent words. Also, although the BRNN slightly outperformed the RNN, it does not appear that the difference in performance is statistically significant.

5.2 NEXT STEPS

This thesis did not address the selection and generation of the words to be connected with the linking morphemes. Therefore, in this is an important problem which remains to be addressed, and it is noted that the prepositional analysis in Sorokin et al. [7] may serve as a useful starting point. For example, whereas 'House *for* dogs' should form a compound like 'doghouse', 'dog *in* house' might form a compound like 'House dog'.

Also, in order to test the efficacy of the model for generation of not yet existing words, an extrinsic evaluation in the form of a user study with native German speakers should be conducted. However, it is outside the scope of the thesis, and will be left for future research.

BIBLIOGRAPHY

[1] S. N. a. H. Ney, "Improving SMT quality with morpho-syntactic analysis.," in *18th Int. Conf. on Computational Linguistics*, 2000.

[2] P. K. a. K. Knight, "Empirical Methods for Compound Splitting," in *EACL '03: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics*, Budapest, 2003.

[3] I. B. Wolfgang Fleishcer, Wortbildung der deutschen Gegenwartssprache, Tubingen: Max Niemeyer Verlag, 1995.

[4] O. V. Q. V. L. Ilya Sutskever, "Sequence to Sequence Learning with Neural Networks," 14 December 2014. [Online]. Available: https://arxiv.org/abs/1409.3215. [Accessed 25 April 2019].

[5] G. Drosdowski, Duden Etymologie: Herkunftswörterbuch der deutschen Sprache, Berlin: Duden Verlag, 1997.

[6] C. Russ, The German Language Today, London: Routledge, 2002.

[7] C. D. a. E. H. Daniil Sorokin, "Multi-label Classification of Semantic Relations in German Nominal Compounds using SVMs," in *CCLCC*, Tubingen, 2014.

[8] J. a. R. T. Goldsmith, "Automatic Collection and Analysis of German Compounds," in *The Computational Treatment of Nominals*, Montreal, Universite de Montreal, 1998, pp. 61-69.

[9] P. Geutner, "Using morphology towards better large-vocabulary speech recognition systems," in *ICASSP*, 1995.

[10] A. P. F. a. P. R.-B. Berton, "Compound Words in Large-Vocabulary German Speech Recognition Systems," in *Proceedings ICSLP 96*, 1995.

[11] F. Cap, *Morphological Processing of Compound Words for Machine Translation*, Stuttgart: Dissertation, Institute for Natural Language Processing (IMS), University of Stuttgart, 2014.

[12] O. T. A. B. S. a. E. D. Vinyalis, "Show and Tell: A Neural Image Caption Generator.," arXiv:1411.4555 [cs.CV], 2014.

[13] P. K. a. K. Knight, "Methods for Compound Splitting," in *Proceedings on the Tenth Conference on European Chapter of the Association for Computational Linguistics*, Budapest, 20013.

[14] S. H. a. J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[15] A. Karpathy, "The Unreasonable Effectiveness of Recurrent Neural Networks," 21 May 2015. [Online]. Available: karpathy.github.io/s015/05/21/rnn-effectiveness. [Accessed 21 May 2019].

[16] K. C. a. Y. B. D. Bahdanau, "Neural Machine Translation By Jointly Learning to Align and Translate.," no. arXic preprint arXiv:1409.0473, 2014, 2014.

Appendix 1: List of Variables and Parameter Values for RNN:

Variable	**Value**
x	31 (length of character dictionary)
y	31 (length of character dictionary)
m	Number of training examples (here 22,337)
a	Length of total output (here 25 set as length of longest GCW in the corpus)
$a^{<t-1>}$	Matrix of size *x* * *m*
$x^{<t>}$ –	(31,m) one-hot vector of input chars
y<t>	(31,m) one hot vector of predicted char
Waa	Matrix of size *a* * *a* (here 25 x 25)
Wax	Matrix of size *a* * *x* (25 x 31)
Way	Matrix of size *a* * *y* (25 x 31)
Wya	Matrix of size *y* * *a* (
by	Vector of length *y* (31,1)
ba	Vector or length *a* (31,1)