University of Helsinki
Department of Digital Humanities
Language Technology

Master's thesis

Sense-aware Unsupervised Machine Translation

Teemu Vahtola
014325632

Supervisors: Jörg Tiedemann, Alessandro Raganato November 3, 2020

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Tiedekunta – Fakultet – Faculty Humanistinen tiedekunta	Koulutusohjelma – Utbildningsprogram – Degree Programme Kielellisen diversiteetin ja digitaalisten ihmistieteiden maisteriohjelma	
Opintosuunta – Studieinriktning – Study Track Kieliteknologia		
Tekijä – Författare – Author Teemu Vahtola		
Työn nimi – Arbetets titel – Title Sense-aware Unsupervised Machine Translation		
Työn laji – Arbetets art – Level Pro gradu	Aika – Datum – Month and year 11/2020	Sivumäärä– Sidoantal – Number of pages 62

Tiivistelmä – Referat – Abstract

Modernit sanaupotusmenetelmät, esimerkiksi Word2vec, eivät mallinna leksikaalista moniselitteisyyttä luottaessaan kunkin sanan mallinnuksen yhden vektorirepresentaation varaan. Näin ollen leksikaalinen moniselitteisyys aiheuttaa ongelmia konekääntimille ja voi johtaa moniselitteisten sanojen käännökset usein harhaan. Työssä tarkastellaan mahdollisuutta mallintaa moniselitteisiä sanoja merkitysupotusmenetelmän (*sense embeddings*) avulla ja hyödynnetään merkitysupotuksia valvomattoman konekäännösohjelman (*unsupervised machine translation*) opetuksessa kieliparilla Englanti-Saksa.

Siinä missä sanaupotusmenetelmät oppivat yhden vektorirepresentaation kullekin sanalle, merkitysupotusmenetelmän avulla voidaan oppia useita representaatioita riippuen aineistosta tunnistettujen merkitysten määrästä. Näin ollen yksi valvomattoman konekääntämisen perusmenetelmistä, sanaupotusten kuvaus joukosta lähde- ja kohdekielten yksikielisiä vektorirepresentaatioita jaettuun kaksikieliseen vektoriavaruuteen, voi tuottaa paremman kuvauksen, jossa moniselitteiset sanat mallintuvat paremmin jaetussa vektoriavaruudessa. Tämä mallinnustapa voi vaikuttaa positiivisesti konekäännösohjelman kykyyn kääntää moniselitteisiä sanoja.
Työssä merkitysupotusmalleja käytetään saneiden alamerkitysten yksiselitteistämiseen, ja tämän myötä jokainen konekäännösmallin opetusaineistossa esiintyvä sane annotoidaan merkitystunnisteella. Näin ollen konekäännösmalli hyödyntää sanaupotusten sijaan merkitysupotuksia oppiessaan kääntämään lähde- ja kohdekielten välillä.

Työssä opetetaan tilastollinen konekäännösmalli käyttäen tavanomaista sanaupotusmenetelmää. Tämän lisäksi opetetaan sekä tilastollinen että neuroverkkokonekäännösmalli käyttäen merkitysupotusmenetelmää. Aineistona työssä käytetään WMT-14 *News Crawl* -aineistoa. Opetettujen mallien tuloksia verrataan aiempaan konekäännöstutkimuksen automaattisessa arvioinnissa hyvin menestyneeseen tilastolliseen konekäännösmalliin. Lisäksi työssä suoritetaan tulosten laadullinen arviointi, jossa keskitytään yksittäisten moniselitteisten sanojen kääntämiseen. Tulokset osoittavat, että käännösmallit voivat hyötyä merkitysupotusmenetelmästä. Tarkasteltujen esimerkkien perusteella merkitysupotusmenetelmää hyödyntävät konekäännösmallit onnistuvat kääntämään moniselitteisiä sanoja sanaupotusmenetelmää hyödyntävää mallia tarkemmin vastaamaan referenssikäännöksissä valittuja käännöksiä. Näin ollen laadullisen arvioinnin kohdistuessa yksittäisten moniselitteisten sanojen kääntämiseen, merkitysupotusmenetelmästä näyttää olevan hyötyä konekäännösmallien opetuksessa.

Avainsanat – Nyckelord – Keywords
konekääntäminen, machine translation, unsupervised machine translation, sense embeddings, machine learning

Säilytyspaikka – Förvaringställe – Where deposited
Keskustakampuksen kirjasto

Muita tietoja – Övriga uppgifter – Additional information

Contents

1 Introduction

Machine translation is one of the prominent tasks in Natural Language Processing. With recent progress in machine learning methods within the field, the systems have gained substantial improvements in performance. Notably incorporating neural network methods into machine translation algorithms has pushed the results even further towards high quality translations. The triumph of machine translation has been highly dependent on large amounts of parallel data that the statistical and neural models have been able to utilise more efficiently compared to the previous paradigms (Bentivogli et al., 2016). The needed amount of parallel data, however, is not available in the majority of languages, which has pushed research into experimenting with unsupervised, monolingual methods, that require no parallel data at all. After Artetxe et al. (2018c), one of the firsts attempts in unsupervised machine translation, showing the potential of monolingual neural machine translation, a lot of research has been done in unsupervised machine translation paradigm. Still, there is plenty of room for improvement when BLEU (Papineni et al., 2002) scores, which measure the translation quality in a range from 0 to 100, the higher being the better, are considered. The current state-of-the-art models gain a BLEU score of 17.43 in German-English and 14.08 in English-German with a statistical model (Artetxe et al., 2018b) trained with the WMT14 data, 25.19 in German-English and 20.23 in English-German with a combination of a neural and a phrase-based statistical model (Lample et al., 2018c) trained with the WMT-16 data or 27.0 in German-English and 22.5 in English-German (Artetxe et al., 2019) also with a hybridization model trained with the WMT-14 data.

The approach in this thesis is not aiming at improving the BLEU scores per se, but in improving the overall quality of translations in a way that is not necessarily measurable with the common automatic evaluation methods. Lexical ambiguity introduces a difficult to solve problem for machine translation. Thus, in this thesis, I investigate the effect of training an unsupervised machine translation model with sense embeddings, instead of typical word embeddings, and focus on how the sense embedding method can help in solving the lexical ambiguity problem by improving the results on word-level translation of ambiguous words. To my knowledge, similar kind of study where word senses are identified directly from the data and used in an unsupervised machine translation pipeline has not been conducted before. For this reason, the results give an insight of a way of possibly improving translation models in general by performing word sense disambiguation as a part of the pipeline. The disambiguation method I use in this thesis utilises similar

methods to typical word embedding training. However, instead of learning one representation for each word in the data, multiple representations are learnt to model each identified meaning of the word. The method can be seen as artificially expanding the vocabulary so that for each word there are multiple different options, i.e. the learnt senses, to choose from. As a result of the word sense disambiguation task, the translation model operates on the meanings of words instead of words, and hypothetically succeeds in translating ambiguous words correctly between English and German.

1.1 Motivation

Current state-of-the-art word embedding systems, such as Word2vec (Mikolov et al., 2013a), GloVe (Pennington et al., 2014) and fastText (Bojanowski et al., 2017), generate vector representations for each word in the given data. However, they make a naive assumption that including only one representation for a word is enough to capture the semantics of language disregarding the fact that language is highly ambiguous. This in turn creates a difficult challenge for machine translation systems which learn different senses for words as part of their end-to-end translation task (Rios Gonzales et al., 2017). As Rios Gonzales et al. (2017) states, choosing a wrong representative of a lexeme may result in wrong or incomprehensible translations due to different senses having different translations which alone confirms the need of word sense disambiguation in machine translation and acts as a motivator to this thesis. A textbook example of a polysemous word in English is *bank*. Multiple meanings exist for the word, and translation should be in accordance with the meaning. Thus, when discussing about the financial institution, one would want the German translation to be *die Bank*, as opposed to for example *das Ufer* which could also be a hypothesis but instead of financial institution, carries the meaning of a shore.

To answer the problem of polysemy in machine translation, instead of learning single-sense word embeddings, multiple vector representations are learnt for each word to capture the different meanings of each word in the training data. As a result, instead of word embeddings, the representations are considered to be sense embeddings that, later on, the translation model operates with. The method of training sense embeddings straight from the data can be seen as an unsupervised word sense disambiguation task, typically referred to as word sense induction, since the aim is to disambiguate the senses automatically from the original input without any pre-set user-annotation of word senses (Bartunov et al., 2016, 1). AdaGram (Bartunov

et al., 2016), an extension to the Skip-gram algorithm (Mikolov et al., 2013a), is used in learning the sense embeddings. Learnt sense embeddings are then used to sense annotate the data that goes into the translation system. As a result of the annotation process, an evaluation of the lexical choices the translation system makes in the translation process is possible by using the corresponding sense embedding model.

In accordance with the previous works in machine translation, English-German language pair is used. One reason to choose this language pair is its high amount of readily usable data, as the language pair is widely used in machine translation research. In addition, an argument advocating the chosen language pair is that conducting the type of research this thesis represents with a widely used language pair with very high resources opens a possibility of future research with lower resource language pairs when the method is first examined with a well represented pair. The main reason for the chosen language pair is to enable a consistent comparison between the proposed and the previous models due to the language pair being widely used in machine translation tasks. Similarly, to ensure comparability, the used data is the well established News Crawl data of the WMT 2014 shared translation task[1].

1.2 Research question and hypothesis

To address the presented problem in machine translation, I am integrating a word sense disambiguation module into a machine translation pipeline to answer the following research question: does integrating a word sense disambiguation module into an unsupervised machine translation pipeline result in better translation quality of ambiguous words in comparison to a current state-of-the-art unsupervised machine translation system and thus increase the adequacy of the translation in general?

My hypothesis is that when the data is sense annotated, the model can better translate ambiguous words because disambiguation could enable better mapping of the disambiguated words into a shared vector space. With word embeddings, only one representation should cover the different meanings of the word. With sense embeddings, each word is represented by multiple embeddings, each having their representation based on their context in the data. The embeddings can be used to annotate the data so that each token is annotated with a sense identifier. Intuitively, each sense is their own token in the input data and the model should learn their differences the same way it distinguishes different unambiguous words in general. As an example, the

[1] `http://www.statmt.org/wmt14/translation-task.html`

word *Gericht* can be translated into English as *dish* or *court*. If the word is represented only by one representation, the translation could be either of the English options and result in a wrong translation. After the annotation process, the data may contain two representations for the word *Gericht*, one that carries the meaning of *dish* and one carrying the meaning of *court*, which hopefully points the model to a correct prediction.

1.3 Structure of the thesis

After the introductory section, this thesis follows the structure presented here. The second chapter focuses on the theoretical framework that I base this thesis on and presents the most important previous research on the matters related to this thesis. The need of incorporating word sense disambiguation in machine translation is motivated and presented in more detail in section 2.1. In section 2.2, I present background to some methods of dealing with ambiguity in Natural Language Processing, especially in machine translation, giving important background to the sense embedding method used in this thesis. Lastly, in section 2.3, background on machine translation and evaluation of machine translation meaningful to the scope of this thesis are presented.

In the third chapter, the methods used in this thesis are presented in detail. Section 3.1 focuses on the training of the sense embeddings for sense annotating the data, while section 3.2 is about the statistical machine translation framework used to train a statistical translation model both with word embeddings and with sense embeddings. In section 3.3 the neural machine translation framework used for the NMT implementation with sense embeddings is presented. Lastly, in subsection 3.4, evaluation methods are discussed.

In the fourth chapter, the experimental setup is presented in detail. The focus is on the used data and the architecture and hyperparameters of the models to ensure the reproducibility of the study.

In chapter five, the results of the experiments are presented and discussed. The results of the sense embedding training are shortly presented and analysed in section 5.1, and the results of the translation task itself are presented in 5.2. Finally, chapter six concludes the thesis.

2 Background & Previous Research

The goal of machine translation is to create systems that are able to automatically translate between languages carrying the meaning of the source into the target fluently and adequately. To be able to obtain a meaningful translation between the source and the target languages, the designed translation systems must have the capability of carrying enough semantic features of the language at hand to the other. In this section, I present the analytical framework surrounding the objectives of my thesis as well as introduce the main literature concerning the topic. In the following subsections, I focus on the theoretical background on the effect of ambiguity on machine translation further motivating the need for this research and present previous research about the topic meaningful to this thesis. I also present different approaches that have been conducted to solve the ambiguity problem in Natural Language Processing focusing on the machine translation field. In addition, I present some of the most important research in statistical and neural machine translation related to this thesis and the methods utilised in this thesis.

2.1 Ambiguity and machine translation

As the end goal of machine translation is the most adequate and fluent translation possible, some features in language introduce the field with difficult to solve problems. Polysemy is one of the bigger problems to solve in order to create high quality translations computationally.

A lot of work has been conducted in the field of general linguistics investigating ambiguity in language, but the work in most relation to this thesis is the research on lexical semantics. Saeed (2015, 51) describes lexical semantics as the investigation of the meaning of each word in a language and the demonstration of how the interrelated use of the words create their meanings. In terms of this thesis, interrelated use of words in a language and ambiguity caused by it is interpreted as representing polysemy. The very much quoted saying of Firth (1957, 11), *you shall know a word by the company it keeps*, is very well suited again. The preceding and the posterior words of the target word at hand define the meaning of the word, and the context of a word is widely used as a defining factor of words' semantics in this thesis, too.

The problem that ambiguity causes in machine translation comes from the fact that many lexemes can carry multiple meanings, i.e. the words can be polysemous. A textbook example of a polysemous word in English is *bank*. WordNet (Fellbaum, 1998), an English lexical database, lists ten different

sense	word	explanation
1	bank	sloping land (especially the slope beside a body of water)
2	bank	a financial institution that accepts deposits and channels the money into lending activities
3	bank	a long ridge or pile
4	bank	an arrangement of similar objects in a row or in tiers
5	bank	a supply or stock held in reserve for future use (especially in emergencies)

Table 1: First five senses of the noun *bank* from WordNet

meanings for the word *bank* in the grammatical category of nouns only. If verbs are also taken into consideration, the total number of senses for *bank* rise to eighteen. Examples of five different senses listed in WordNet are given in table 1.

Making a wrong decision of the chosen sense while translating into another language might result in awkward and even incomprehensible translations where a word appears in a completely incoherent context. As the commonly used word embedding algorithms such as Word2Vec (Mikolov et al., 2013a) only model one representation for each word, all eighteen senses of the given example would be represented in the same n-dimensional vector. Even though the models have been shown to carry semantic properties (Mikolov et al., 2013d), the most frequent sense of the word dominates the representation or the senses are mixed (Bartunov et al., 2016). It is a very strong assumption that one representation for a word would be able to represent the ambiguous nature characteristic to natural languages. In the case of machine translation, if the embedding spaces of the source and the target language would be perfectly isometric, perhaps single-sense embeddings could be enough. However, the same level of ambiguity is not present between languages and, as a result, the embedding spaces will not be perfectly isometric. Thus, a machine translation system could choose a wrong word from the target language as a translation when ambiguous words are represented as one representation in the embedding space. As an example from Finnish, the word *kuusi* could translate to English as the number six, as pine or as your moon. As a result, modeling only one representation for the word in Finnish could result in the wrong translation since the words are highly dependent on the context and one even demands a personal pronoun in addition to the noun in the translation. Thus, dealing with polysemy is needed to generate better translation of ambiguous words and as a result improve the quality of machine translation in general. Many approaches have been proposed to

deal with the ambiguity problem, and I will present the most related ones to my work next.

2.2 Towards unambiguous NLP

A lot of work has been conducted in the Natural Language Processing (NLP) community to deal with ambiguity in language. As this thesis focuses on embedding models in machine translation, I will shortly present some background to word embeddings, which per se do not serve as an answer to the ambiguity problem, but do work as a starting point for more fine-tuned solutions to model lexical ambiguity in machine translation and NLP in general. In the following sections, I will shortly present background to word embeddings as well as show how embedding models are utilised to model meaning.

2.2.1 Words as vectors

Recent improvements in NLP are highly dependent on the effect of high quality word embeddings. The intuition behind the embedding models lays in the distributional hypothesis (Harris, 1954) which states that words that appear in a similar context tend to be semantically similar. Even though word embeddings as such are not an answer to the ambiguity problem, they have shown to be a powerful method in NLP in representing linguistic properties. They also work as a background to the meaning representation method used in this thesis.

The development of machine learning methods, especially neural networks, has lead into the possibility of representing words as dense vectors in vector space. Such neural network based models as Word2Vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) have outperformed previous n-gram models in nearly every downstream task. The modern neural network based models are basically neural networks that represent words as vectors, and cluster similar vectors close to each other in the embedding space. More related than GloVe to this work is Word2Vec, which includes two algorithms: Skip-gram and Continuous Bag of Words (CBOW). In the case of the Word2Vec algorithms, they take an input of words and perform a binary classification task of predicting either context words based on the given target word or target word based on the given context. The hidden state vectors of the classifier are saved after the training has reached convergence and taken as representations of the words in the data.

As the words are represented as vectors in the vector space, the distribu-

tional hypothesis is intuitively present in the review of the embeddings. Vectors being elements of direction and magnitude, they can be visualized as lines in space where each line represents a word in the vocabulary. When positioned in the embedding space their similarities can be measured using Euclidean distance or cosine similarity. Examples show that the embeddings are so powerful a method that they can carry syntactic and semantic information. A popular example is presented in Mikolov et al. (2013d), where the authors show that the models are capable of rather successfully answering analogy questions *a:b c:d*, where d is unknown, by finding the word embeddings for a, b and c, computing $y = vec_b - vec_a + vec_c$, calculating the cosine similarity between word embeddings in the model and y and choosing the word vector with the highest similarity score as d. Popular example of the analogy question is from Mikolov et al. (2013d): taking the vector representations of words *king*, *man*, and *woman* and calculating the result of $vec_{king} - vec_{man} + vec_{woman}$ results in a vector that is very similar to the representation of *queen*. Rather successfully performing shown kind of analogy questions is characteristic to other low dimensional word embedding models, too.

Because of their inherent ability to carry syntactic and semantic information, word embeddings have been utilised in many NLP downstream tasks, machine translation being one. Presented in Mikolov et al. (2013b), dictionaries and phrase tables used in statistical machine translation can be improved from distributed representations by performing a linear transformation between monolingual embedding spaces using a small seed dictionary. The linear mapping works because the monolingual embedding spaces are similar in different languages, interestingly, but not very surprisingly, even between distant language pairs such as English and Vietnamese (Mikolov et al., 2013b). Similar methods have since been applied into unsupervised machine translation paradigm (Artetxe et al., 2018a; Lample et al., 2018b).

Accepting the distributional hypothesis, similar vector spaces occur in different languages because similar words tend to occur in similar contexts. According to Mikolov et al. (2013b, 1): *all common languages share concepts that are grounded in the real world*, which as a result leads into vectors positioning similarly in the vector spaces. Mikolov et al. (2013b) show that vectors that represent numbers and certain animals are very similarly arranged in their respective vector spaces in English and Spanish. As has been noted before, the embedding models only model one representation that is supposed to represent every sense of a given word. The limitation of not being able to differentiate different meanings of a word is referred to as *meaning conflation deficiency* in Camacho-Collados and Pilehvar (2018). This limitation may

result to many unwanted problems in machine translation, as was noted in section 2.1. Letting the more frequent meanings dominate the representation of words is problematic as is, but more so when Zipf's law is considered: the more frequent a word is, the more meanings it has (Camacho-Collados and Pilehvar, 2018, 744). The same conclusion was also made in Bartunov et al. (2016) where it was shown in practice that more frequent words carry more senses compared to the less frequent ones. Yaghoobzadeh and Schütze (2016) show that a single representation for a word can effectively represent multiple meanings when all the meanings are frequent enough. However, Yaghoobzadeh and Schütze also notice that the frequent senses dominate the representation leaving the rarer ones unnoticed. For being able to translate even the rarer senses correctly, learning sense embeddings instead of word embeddings is well justified. A lot of work has been conducted in representing the meaning of a word in vector space, and the most related ones to this work are presented next.

2.2.2 Vector representations of meaning

Low-dimensional word embeddings have been utilised a lot in different NLP tasks and significant improvement has been reported also in machine translation after integrating word embeddings into the pipeline (Camacho-Collados and Pilehvar, 2018). In spite of having shown to capture semantic and syntactic properties of language, word embeddings have one significant limitation: they only represent one meaning for a word. The solution to the ambiguity problem used in this work is to build multiple representations for each word representing each sense inducted from the corpus, i.e. creating sense embeddings instead of typical word embeddings. Different approaches in learning sense embeddings are divided into unsupervised and knowledge-based methods. In unsupervised approaches the sense distinctions are induced from text corpora alone, while knowledge-based methods utilise an external sense inventories. In this thesis, I focus on unsupervised methods, so the knowledge-based methods are described only briefly.

Knowledge-based methods use an external sense inventories in creating representations of senses. Inventories such as WordNet or BabelNet (Navigli and Ponzetto, 2012) can be used for example to guide the learning process as in Iacobacci and Navigli (2019), where the proposed model learns word and sense embeddings and uses pretrained embeddings from BabelNet as an objective to the learning process to use the semantic information of the knowledge base as an extra information to steer the model. Very similarly to the method proposed in this thesis, Iacobacci et al. (2015) first annotate

their corpus with sense annotations and then learn word embeddings of the annotated corpora with Word2Vec to get sense embeddings. However, while Iacobacci et al. use a knowledge resource to annotate the corpus, the system used in this thesis is unsupervised, meaning it learns the senses straight from the text corpus.

As distributed word embeddings have been shown to be able to model similarity of words, they have also been highly beneficial for learning sense-specific representations. The first multi-sense model to build on top of the famous word embeddings algorithms is Multiple-Sense Skip-Gram (MSSG) by Neelakantan et al. (2014) which extends Skip-gram by maintaining multiple representations for each word based on the meanings. The proposed system of Neelakantan et al. takes the average of a word's context vectors as a representation of the context which are then clustered. The sense of a word is acquired by taking the closest cluster to the context representation of the word. Neelakantan et al. (2014), Liu et al. (2015a), Liu et al. (2015b), and Nguyen et al. (2017) have experimented with topic modeling to learn multiple topic embeddings for a word so that each meaning of a word is defined by the topic. According to Camacho-Collados and Pilehvar (2018), the problem with the previous joint models, i.e. models that simultaneously induce the senses and perform the representation learning, is that they can only learn a fixed number of senses. While the large majority of words are monosemous, ambiguous words are more likely to occur in a real text than their proportion suggests (Camacho-Collados and Pilehvar, 2018, 753). Thus, assigning a number of senses to learn from might not be the optimal manner in learning sense embeddings. However, sometimes limitations are needed in order to maintain a feasible vocabulary size as the induced senses can easily explode the size of the vocabulary.

Bartunov et al. (2016) answer the problem of fixed senses with their model, that is also built on Skip-gram, but utilises stick-breaking (Sethuraman, 1994) as a part of Dirichlet process (Ferguson, 1973) to be able to learn practically unlimited number of senses for each word. In this thesis, I use the model by Bartunov et al. (2016) to learn sense embeddings directly from the corpus.

Another problem Camacho-Collados and Pilehvar (2018, 753–754) point out is that learning sense embeddings based on the context but not disambiguating the context means that the senses are conditioned on the word embeddings, not the sense embeddings of the context. AdaGram (Bartunov et al., 2016) also falls into this category of models. However, Bartunov et al. (2016, 3) point out that using meanings of the input words to predict the context words is enough to answer the ambiguity problem, at least so that the

complexity of training a context meaning-aware model would not be worth it.

2.3 Machine Translation

Sense embeddings have been evaluated in tasks such as word prediction and word sense induction, but their effect to machine translation has not been properly evaluated. Thus, utilising sense embeddings in unsupervised machine translation is the novelty of this thesis. My proposed model is based on the statistical model presented in Artetxe et al. (2018b) due to its notable BLEU scores. I use the model as my baseline on which I build my proposed model with sense embeddings. While the paper by Artetxe et al. (2018b) acts as the biggest influence, another experiment is conducted in the continuum of unsupervised machine translation method demonstrated in Artetxe et al. (2019) which shows the potential of a combination of unsupervised statistical and neural machine translation. In the following sections, I present some background on the machine translation paradigms used in this thesis. The focus of this thesis is on unsupervised methods, but a brief review of statistical and neural models in general is given.

2.3.1 Statistical Machine Translation

Statistical machine translation (SMT) has existed for decades. As the model used in this thesis is unsupervised phrase-based model, I focus on the literature of this paradigm in this section, and leave the historical perspectives as well as the word or syntax-based models aside. In this section, I shortly present the most meaningful previous research in the scope of this thesis and the models I use.

Och et al. (1999) extend the word-based alignment models into an alignment of phrases and the individual words included in the phrases to consider changes in word order between languages as well as one-to-many constructions. Marcu and Wong (2002) experiment with a translation model that can learn corresponding translations between phrases in source and target languages, and do it jointly, i.e. map the source and target sentences simultaneously. Koehn et al. (2003) investigated the usability of different phrase-based translation methods with evaluating different methods with a Bayesian phrase translation model, and presented a combination of scoring functions for extracting phrase tables that have been since present in most statistical models (Artetxe et al., 2018b). The scoring models typically found in

an SMT system listed in Artetxe et al. (2018b) include a phrase table that saves n-grams in the source language and their possible translations in the target language, a language model that calculates the probabilities of word sequences in the target language, a word reordering model to deal with different word orders in source and target languages, and a model for word and phrase penalties which handles the length of the translation sentences. An objective to training is then to maximize a scoring function so that it optimizes the weights of the complete model and maximizes a suitable evaluation metric, typically BLEU in machine translation (Artetxe et al., 2018b, 2–3).

Even though the neural models have lately shown their significant potential in machine translation, statistical models are better suited for certain situations such as lower resource scenarios, out of domain translation, translating rare words, and translating long sentences (Koehn and Knowles, 2017). Even though the mentioned challenges are not necessarily tested in this thesis, increasing the vocabulary with sense annotations lowers the frequencies of the words since each individual word occurrence that would be counted as an example of a word is now considered as an example of a certain meaning of the word. Thus, statistical models' superiority in translating rare words could theoretically have a meaningful impact on the results. In addition, the dataset used in this thesis, the WMT newswire, consists of rather long sentences averaging to approximately 30 words per sentence (Koehn and Knowles, 2017). It must be noted that the set-up in this thesis is by no means low resource the language pair being English-German.

Bentivogli et al. (2016) conduct a careful comparison between statistical and neural translation models in English-German language pair by using a post-edit metric TER (Snover et al., 2006), which measures how much editing has to be done to the translations by a human after translating, finding out neural models perform significantly better on the language pair. The findings would suggest to experiment with a neural machine translation system in this thesis. However, in the unsupervised machine translation paradigm that the models experimented with in thesis represent, the statistical model of Artetxe et al. (2018b) notably scored higher than the neural model of Artetxe et al. (2018c) evaluated with BLEU.

The initial idea for this thesis was to test sense embeddings in unsupervised neural machine translation, but acknowledging the limitations of neural models as well as the notable results of Artetxe et al. (2018b) in the unsupervised paradigm lead the thesis to shift towards SMT however including its original interest in neural translation containing experiments with both paradigms. In the next subsection, I will shortly present the most meaningful research

on NMT for this thesis.

2.3.2 Neural Machine Translation

The objective of this thesis is to utilise sense embeddings instead of typical word embeddings in machine translation. First, I train a statistical machine translation model with sense annotated data, which is the first experiment. The second method experimented is a combination of unsupervised SMT and NMT paradigms. The back-translations of the SMT model are saved and used as a synthetic parallel data in training an NMT model. Because of the experiments with NMT paradigm, I shortly present the most meaningful previous research in supervised NMT to this thesis in this section.

From the 1990s the main paradigm of machine translation has been statistical and as a paradigm it was not superseded until 2014 when Cho et al. (2014b) published their paper experimenting with neural networks in statistical machine translation and presenting the Recurrent Neural Network (RNN) encoder-decoder architecture which they use for scoring phrase pairs in a phrase table to improve the translation quality. An encoder-decoder model uses one neural network to encode the input sequence into a vector representation, which another neural network then decodes into a natural language output. The idea of utilising a similar kind of encoder-decoder architecture in translation was already discussed in Kalchbrenner and Blunsom (2013), however, instead of a RNN model, they use a convolutional n-gram model in the encoder and a RNN in the decoder (Cho et al., 2014b).

Similarly to Cho et al. (2014b), Sutskever et al. (2014) use an encoder-decoder model but unlike Cho et al., Sutskever et al. use Long short-term memory (LSTM) units to encode the input sequence into a vector representation and another LSTM unit to *directly* decode the representation into the target sequence. In Cho et al. (2014a) the authors show that translating long sentences proves to be a difficult task for encoder-decoder models. Bahdanau et al. (2015b) present an approach to answer to the problem of poor performance with long sentences by integrating an attention module into the encoder-decoder system. Building on top of these works, Wu et al. (2016) published their deep LSTM model with attention mechanism reaching notable performance on several language pairs. After the work of Wu et al., NMT has become the major paradigm in machine translation however not completely superseding statistical models. Respectively, neural network based methods exceeded the performance of statistical methods in nearly every metric (Bentivogli et al., 2016). Notably, Vaswani et al. (2017) published Transformer, a state-

of-the-art model relying only on multiple attention cells. In the next section, I shortly present previous research on the unsupervised machine translation paradigm.

2.3.3 Unsupervised Machine Translation

While substantial results have been obtained with modern NMT models, utilising neural networks has only been possible for high resource language pairs. Neural models need massive parallel datasets to work well and the number of parallel sentences has to be in millions. The democratization of machine translation and NLP in general is highly dependent on the possibility to effectively utilise small and lower-quality datasets. Of course, not many languages have such massive parallel datasets available that are needed for NMT, so to make machine translation more applicable, learning algorithms need to be able to utilize monolingual data (Lample et al., 2018a). The first experiment with unsupervised NMT is presented in the paper by Artetxe et al. (2018c), where the authors introduce a completely unsupervised approach to neural machine translation building their model utilising back-translation (Sennrich et al., 2016a) in training their encode-decoder model with monolingual corpora only. In general, unsupervised machine translation is made possible by using linear transformation to map monolingual word embeddings into a shared vector space where semantically similar words are supposed to occur in close proximity. Artetxe et al. (2018c) use the cross-lingual word embedding mapping method in order to be able to train a shared encoder for both translation directions to create language-independent representations of the input that the decoders then transform into the correct language and output (Artetxe et al., 2018c, 4). The model is further improved with back-translation (Sennrich et al., 2016a) which generates a synthetic parallel corpus by translating the input sequence into the source sequence and then training a system in the opposite direction using the synthetic data. The process is continued until a certain training criterion is met resulting in a parallel data with back-translated sentences on one side and original sentences on the other. Since Artetxe et al. (2018c), experiments in unsupervised machine translation have been made both in statistical and neural paradigms. Artetxe et al. (2018b) surpassed the neural model of Artetxe et al. (2018c) with a statistical approach, and Lample et al. (2018c) trained both a phrase-based statistical model and a neural network model to further improve the BLEU scores of unsupervised machine translation respectively.

Artetxe et al. (2019) point out that while SMT provides a better method for initialization between source and target languages, NMT suits better for

the actual translation; thus, building a *hybridization* of the two could be the optimal solution. Similar results are obtained in Lample et al. (2018c), where the authors increase the back-translated data of their NMT system by the phrase-based SMT generated data. Building NMT on top of SMT was already utilised in the supervised paradigm (Kalchbrenner and Blunsom, 2013; Cho et al., 2014b) but using the method in the unsupervised paradigm yields notable results and is the most related to the method I use in this thesis. The hybridization model of Artetxe et al. (2019) as well as the combined model of Lample et al. (2018c) improve the results of unsupervised MT and thus serve as a point of interest to investigate further but with sense embeddings. While in Artetxe et al. (2019) the subword model by Sennrich et al. (2016b) is used as the NMT model to perform NMT hybridization, I use OpenNMT's (Klein et al., 2017) OpenNMT-py framework to train the hybridization model having the NMT model presented in Bahdanau et al. (2015a) as my inspiration.

2.3.4 Evaluation

Evaluation of machine translation is conducted either by a human or by automatic measures. Human evaluation is expensive and slow, whereas automating the task is fast and thus more suitable for the needs of MT research, since fast evaluation of changes in a model is needed to guide the development process. A popular choice for automatic evaluation is BLEU (Papineni et al., 2002), which counts the number of corresponding n-gram matches between the translated sentence and the reference sentences and ranks the translations. The evaluation method captures fluency and adequacy, according to Papineni et al. (2002, 313), because a high rank of matching unigrams corresponds to adequacy, while fluency is obtained by matching longer n-grams cross-linguistically.

As the main focus of the evaluation is not the combination of adequacy and fluency of the output as such but rather the capability of the model to correctly translate ambiguous words, BLEU is not the most optimal evaluation measure. Automatic evaluation metrics do not provide such qualitative analysis needed to carefully evaluate the task (Rios Gonzales et al., 2017). Even though BLEU presents a measurement of fluency and adequacy of a translation, it lacks the capability of evaluating ambiguity in terms of polysemy, and as such does not give a satisfactory result to the research question I aim at answering in this thesis. However, BLEU is the standard evaluation measure for machine translation models, and following a common practice, it is used in evaluating the baseline model and the sense embedding models

I experiment with because it also provides an easy and a reliable comparison between this and earlier works.

As discussed, automatic evaluation measures do not provide a sufficient measurement method for the needs of this thesis, and thus, the proposed models' word-level results must be evaluated otherwise. The presented automatic evaluation method still provides a way for a general evaluation of the models. This section serves as a general background to evaluation in machine translation, and the evaluation methods used in this thesis are further discussed in section 3.4.

3 Methods

In this section, I present the methods I use to train and evaluate the proposed translation models. The objective of the thesis is to test and evaluate the effect of integrating a word sense disambiguation (WSD) module into a machine translation pipeline to investigate whether integrating such module improves word-level translations of ambiguous words in the chosen language pair, English-German, and consequently serve as a method to improve machine translation results in general. The pipeline of my proposed model goes as follows: first, a sense embedding model is trained for both languages, the trained sense embeddings are then used to perform WSD on the data set that is used for training the translation model. The intention is to annotate each token with a sense identifier which is the index of the recognized sense from the sense embedding model. The procedure results in data where each occurrence of a token is actually an occurrence of the meaning of the given token. After annotating the tokens with their identified senses, I use the annotated monolingual data in a statistical machine translation (SMT) framework that includes an algorithm for mapping the monolingual sense embeddings into a shared vector space in order to be able to train an unsupervised SMT model with sense embeddings. Neural machine translation (NMT) has been shown to perform better in the translation task (Artetxe et al., 2019), thus, in addition to training the SMT model, I save the back-translations of both the source and the target language from the last iteration of the iterative back-translation procedure of the SMT training in order to train an NMT model on top of the statistical one to investigate the results of NMT built on top of an SMT model.

All in all, I train two SMT models: one with word embeddings to serve as a baseline and a comparison to my proposed model, and one utilising the sense embedding method. In addition to the SMT models, I train an NMT model with the back-translations from the sense-aware SMT model. The baseline model is trained with traditional word embeddings following Artetxe et al. (2018b), but the second SMT model is trained utilising sense embeddings to serve as a comparison between the statistical and the neural sense embedding-based translation models and the word embedding-based one.

In the following subsections, I present the framework I use for training the sense embeddings as well as the statistical and neural machine translation frameworks I am using. A short discussion about the evaluation methods is also included. The model details such as hyperparameters, as well as results and the evaluation metrics are presented and analysed later in sections 4–5.

3.1 AdaGram: training sense embeddings

In order to perform word sense disambiguation (WSD), some form of sense inventory is needed. As noted in section 2.2.2, different options for performing WSD include using a knowledge-based sense inventory such as WordNet (Fellbaum, 1998) or learning the senses directly from the text. In this thesis I test and evaluate unsupervised methods in sense-aware machine translation, thus, I do not use a knowledge-based sense inventory, but instead train a model to induce senses directly from the raw input. To do this, I use the Adaptive Skip-gram (AdaGram) algorithm presented in Bartunov et al. (2016) to learn the monolingual sense embedding models for English and German. In this section, I shortly introduce the AdaGram model using Bartunov et al. (2016) as the main reference.

AdaGram is a Bayesian nonparametric model built on the Skip-gram algorithm (Bartunov et al., 2016). It uses the Dirichlet process (Ferguson, 1973) with stick-breaking (Sethuraman, 1994) to be able to learn practically an infinite number of senses for a target word adaptively from the data by changing the finite dimensionality of the Bayesian model to infinite dimensions. This way, the number of learnt senses can increase as the introduced data grows. As language is highly ambiguous, in many situations the number of possible senses can not be known *a priori* as the number of possible senses can diverse from one to dozens. In the context of modeling natural languages, naturally, more senses are recognized when the data introduced to the model increases. According to Bartunov et al. (2016, 3), Dirichlet process is a natural choice for situations where the number of possible clusters can not be known beforehand. The same way as Skip-gram uses the center word to predict the context words, AdaGram uses the learnt sense of the center word to predict the context words. More precisely, similarly to how the Skip-gram algorithm uses the target word in the hierarchical softmax (Mnih and Hinton, 2009), AdaGram utilises the index of the induced sense in the softmax function to create the prediction. This way the prediction is dependant on the meaning of the word at hand. (Bartunov et al., 2016, 3.) After the training of the Skip-gram model has reached convergence, the hidden state vectors can be saved and utilised as representations of the words. Similarly, the hidden state vectors learnt by AdaGram can be interpreted as representations of the different senses of a given word.

Precisely, the Skip-gram model aims at maximizing the log probability:

$$\sum_{t=1}^{T} \sum_{c \in C_t} \log p(w_c|w_t), \tag{1}$$

where each context word w_c is sampled from a set C_t of preceding and posterior indices of the target word (Bojanowski et al., 2017). Due to the complexity constraints of the softmax function, $O(n)$, Skip-gram is typically trained using the hierarchical softmax (Bartunov et al., 2016):

$$p(v|w, \theta) = \sum_{n \in path(v)} \sigma(ch(n)in_w \cdot out_n), \tag{2}$$

where θ represents vector representations of all words in the data (Bartunov et al., 2016, 2). In equation 2, Huffman encoding is also utilised. Each word in the vocabulary is a leaf in a binary tree, and each node represents the relative probabilities of its child nodes (Mikolov et al., 2013c). Utilising the hierarchical softmax, instead of updating all the words in the vocabulary, only a portion of words need to be updated, making the computational complexity of the algorithm practically $O(\log n)$ instead of $O(n)$ (Mikolov et al., 2013c).

However, Skip-gram only learns one meaning for each word in the data. For translational purposes this can be insufficient, as suggested by the motivation of this thesis. Thus, AdaGram aims at creating high quality multi-prototype representations, i.e. modeling each sense of an occurring word in the data. Since the number of senses for a word can not be known beforehand, an adaptive approach is necessary. To predict the representation for each sense, the objective of Skip-gram is changed to:

$$p(v|z, k, w, \theta) = \sum_{n \in path(v)} \sigma(ch(n)in_{wk} \cdot out_n), \tag{3}$$

where z represents the index of the active meaning at hand and k the k-th meaning of the word (Bartunov et al., 2016). Thus the classification model is practically changed from predicting the context word based on the target word to predicting the context word based on the *meaning* of the target word (Bartunov et al., 2016, 3). Because of the adaptive nature of the model, the number of learnt senses could be infinite. Whilst not knowing the number of possible senses in advance, not setting any threshold for the number of learnt senses may not be optimal. Bartunov et al. (2016) state that learning more prototypes for a word leads to the senses having more specific meanings, however, they point out, too many learnt prototypes per word

leads into difficulties in interpreting the senses and even creates overlapping senses. Some degree of overlap is also noticeable in my practical experiments with the AdaGram model. I do not go into presenting the hyperparameters in this section, however, but focus on the theoretical display of the model. Information about the hyperparameters in learning the sense embeddings is presented in section 4, where I present the experimental setup as a whole.

After successful training, the sense embedding model can be used similarly to typical word embedding models. AdaGram framework contains a function for performing word sense disambiguation. The function calculates the predictive probability of a meaning, and the number of learnt prototypes after which it calculates the probability for each meaning of a word in a given context. The prediction is then calculated as the posterior probability over the context words given the input word. (Bartunov et al., 2016, 5.) I use this provided function in this work in performing WSD on the data.

In order to operate on sense embeddings in the machine translation pipeline, I perform WSD on the data that I use as an input to my translation model. Each token in the data is disambiguated with respect to its context, and added an identifier that represents the induced sense of the token. This way, when in the translation pipeline the model learns word embeddings from the data, the model in fact works with sense embeddings instead of word embeddings as each token really represents a sense of the word. Instead of having one possible prototype for a word as in typical word embedding models, the model may now include multiple representations. The resulting embeddings should be able to perform the same tasks as typical word embeddings, but hypothetically carry more semantic features than single-sense word embeddings. A deeper look into the learnt sense embeddings is presented in section 5.1. In the next section, I shortly present the framework that I use for training the baseline statistical machine translation model as well as the sense-aware statistical machine translation model.

3.2 Monoses: SMT framework

I use Monoses, the unsupervised statistical machine translation (SMT) system presented in Artetxe et al. (2018b), as a main component for three experiments:

1. I create a baseline model on which I can fairly compare my model to.
2. I train a statistical machine translation model utilising sense embeddings.

3. I save the back-translations from the iterative back-translation process in order to use them as an artificial parallel data for training a neural machine translation (NMT) model.

The first SMT model serves as a fair baseline in comparison between the original Monoses system and my method utilising the sense embeddings. The NMT model is an attempt to build an unsupervised NMT system on top of the Monoses model, which is a similar approach to the one presented for example in Artetxe et al. (2019).

Monoses model uses the Skip-gram algorithm with negative sampling to learn word embeddings. To model compositional phrases, Monoses also learns n-gram embeddings for phrases longer than one token. As the original Skip-gram algorithm predicts the context words in a given window based on the target word, the n-gram model looks at all the n-grams of different length in the window pairing them with the current context word and updates the hidden state values of the n-grams without updating the values of the given context. Thus, the word embeddings are trained with Skip-gram as usual, and training the n-gram embeddings does not interfere with training the word embeddings and, as a result, the model learns word embeddings as well as n-gram embeddings (Artetxe et al., 2018b). As the input to Monoses is monolingual, the resulting embedding models of the Skip-gram procedure are also monolingual. In order to utilise the embeddings in translation task, the different monolingual embeddings are mapped into a shared vector space. The intuition is that a word with a similar meaning appears with a similar context in different languages. Thus, the words, or n-grams, should be positioned rather similarly in the different monolingual vector spaces. The cross-lingual word embedding mapping is integrated into the Monoses pipeline with VecMap, an algorithm described in Artetxe et al. (2018a).

VecMap performs a linear transformation on the vector representations in the two monolingual spaces to match the corresponding embeddings in each language to a shared vector space (Artetxe et al., 2018a). In the case of the proposed model, the data is sense annotated, so the mapping should align the different senses of each word accordingly; as an example, the English word *bank* with the sense of *river bank* should not be aligned with the German word *Bank*, the financial institution, but rather with *Ufer*, i.e. the riverbank sense. As I have performed WSD and tagged my input data with sense annotations, the vocabulary is artificially increased. Thus, the data could include two senses for the English word *bank*: *bank*0 and *bank*1, where *bank*0 is a representation of the sense *financial institution* while *bank*1 is a representation of the sense *river bank*. For Monoses, and thus also the integ-

rated VecMap algorithm, these instances are two different tokens. For this reason, theoretically, the mapping should not be any different from mapping between naturally unambiguous words. Evaluation of the linear transformation in the VecMap procedure is presented in Artetxe et al. (2018a) where it is shown that the method gains the state-or-the-art results in unsupervised cross-lingual word embedding mapping.

In order to perform the cross-lingual mapping of the monolingual embeddings, VecMap performs four steps: it normalizes the embeddings, creates an initialization of the mapped embeddings which it then iteratively improves, and finally performs symmetric re-weighting to further improve the quality of the mapping (Artetxe et al., 2018a). Length normalization is a rather typical step when performing similarity calculations of vectors as normalizing the length of the vectors enables calculating the distances between two vectors with dot product. After the embeddings are mean centered and length normalized, the similarity of two embeddings can be measured with dot product as the dot product is equivalent to the cosine similarity of the given embeddings (Artetxe et al., 2018a, 791).

VecMap algorithm creates an initialization of the mapping which is improved in the later steps. The initialization relies on a strong assumption that the two embedding spaces are isometric. If the embedding spaces would be perfectly isometric, the representations of two equivalent words, or n-grams, in different languages would be exactly the same when the vectors are sorted independently. The translation to a word or a corresponding n-gram could then be acquired by nearest neighbour search picking the one with the maximum closeness value. The embedding spaces in question are square roots of the similarity matrices acquired by singular value decomposition from the original embedding matrices. The resulting similarity matrices are aligned and used in building an initialization of the cross-lingual embedding mapping after normalization. (Artetxe et al., 2018a, 791–792.) The self-learning procedure that actually maps the embeddings presented in Artetxe et al. (2018a) is built on the similar proposition presented in Artetxe et al. (2017). Basically, the mapping is conducted in two steps: computing the optimal mapping matrix that minimizes the distances of the embeddings for the dictionary entries, and building the optimal dictionary by using nearest neighbour search from one language to the other (Artetxe et al., 2018a, 2017). When the process is iterated until convergence, the dictionary gets better and better with each iteration. To further improve the results of the cross-lingual word embedding mapping, VecMap inducts the dictionary stochastically, considers only k most frequent words in each language, uses Cross-domain Similarity Local Scaling (Lample et al., 2018b), and induces the dictionary in bidirectionally

(Artetxe et al., 2018a). Finally, after the described self-learning has found the optimal solution, VecMap performs symmetric re-weighting in both languages to finalize the cross-lingual embedding mapping.

After learning the cross-lingual word embedding mapping, Monoses procedure moves into inducing the phrase table. Monoses induces the phrase table by calculating the softmax of the cosine similarities of two embeddings, both unigram and n-gram embeddings, iteratively over all target language embeddings to find the corresponding target embedding for the given source embedding (Artetxe et al., 2018b). The results are then improved with iterative back-translation (Sennrich et al., 2016a). Intuitively, back-translation iterates over the monolingual corpus in one direction, and trains another system on the results to work in the other direction. This way, the model improves itself bidirectionally on each iteration until convergence. After the training process is finished, the model can be used in translation.

3.3 OpenNMT: NMT framework

In addition to creating a sense-aware SMT model and comparing that to a Monoses baseline trained with word embeddings, I use the back-translations acquired from the final iteration of the iterative back-translation process of the SMT pipeline to train a neural machine translation model with the back-translations. The method has been shown effective for example in Artetxe et al. (2019) but has not been tested with sense-aware system.

For the NMT implementation, I use OpenNMT (Klein et al., 2017), an open-source NMT framework aimed to serve as an easy way to build implementations of NMT models (Klein et al., 2017). The OpenNMT framework includes a sequence-to-sequence model that utilises recurrent neural networks and also provides support for the different options needed for a state-of-the-art NMT system (Klein et al., 2017). The actual design of the used NMT architecture, i.e. inclusion of different technologies such as attention, is left for the end-user, which makes OpenNMT an optimal tool for fast implementation of different neural machine translation models. The exact model details and hyper-parameters that I use in the NMT training are presented in section 4.4. In general, the model is an encoder-decoder model with a bidirectional recurrent neural network with long short-term memory (LSTM) on the encoder side and a recurrent neural network with LSTM on the decoder side.

3.4 Evaluation methods

While the meaningful part of the evaluation is word-level translations of ambiguous words, a more general evaluation is performed with *multi-bleu.perl* script of the Moses package to get the BLEU scores. Results are acquired for all the models including the model trained with word embeddings, the statistical model trained with sense embeddings, and the hybridization model trained with the back-translations. As noted in section 2.3.4, the automatic evaluation tools do not cover all the necessary features to evaluate the models' capability of dealing with ambiguity. Therefore, the performance of the sense embedding-based systems is also evaluated qualitatively. For this, I randomly sample a small set of sentences from the data, and analyse how the translation models performed compared to the reference translation. I look into the overall quality of the translation as well as the selected senses of the words. In addition, I selectively sample a set of potentially ambiguous words from each language and compare the translations of the words in their context to investigate how the model succeeds in translating ambiguous words. I retrieve the nearest neighbours of the chosen sense to find out which sense cluster is chosen in the given context, and consider the context to evaluate whether the chosen sense is the correct one. I evaluate the same examples in all models to see whether there is any difference in the models based on these examples. The translation results are presented in section 5.2.

In addition to evaluating the translations, an investigation of how the sense embeddings work is needed in order to make sure the sense annotation is correct. For this, I collect a sample of possibly ambiguous words, and compare their nearest neighbours to see how the sense embedding models cluster semantically similar words. A mere comparison of the nearest neighbours does not act as a thorough evaluation, but it provides enough insight in the scope of this work to evaluate the rationality of the sense embedding models. The small sample of possibly ambiguous English words contains one word from the noun class, one from the adjective class and one from the pronoun class, sampled selectively by the author. Random sampling of the examples could have been performed. However, as the meaning of the sampling is only to perform a sanity check on the learnt sense clusters, and a more thorough evaluation of the sense embedding method is presented in Bartunov et al. (2016), selective sampling is justified. The noun used in the analysis is the typical example of English polysemous word: *bank*, and was taken as an example because of its nature as a textbook example when polysemy in English is discussed. The sampled adjective is *atomic*, and was chosen merely because it appeared in the first few lines of the document, and the data included five

different senses of the word. The sampled pronoun is *it*, and was selected as an example of the closed word class which should not be too ambiguous but interestingly has five learnt prototypes in the model. The nearest neighbours of the different senses of the words are retrieved and analysed for an insight of the sense clusters. In addition, the senses are observed in the context they appear in in the data to further investigate the suitability of the cluster. In addition to the English words, one potentially polysemous German noun is investigated in a similar manner to see whether the model also learns to distinct meanings of German words. The results are presented and discussed in section 5.1.

4 Experimental setup

In this section, I present the experimental setup of this thesis. First, I discuss the data I use in the experiments. Next, I present the preprocessing procedure that I perform on the data. In section 4.2, I present the parameters used in learning the sense representations, after which I show the parameters used in the SMT training. Last, I present the model details of the NMT model used for building the NMT model on top of the SMT model.

4.1 Dataset

As training data, I use the WMT 2014 shared translation task's monolingual English and German News Crawl 2007–2013 datasets. The datasets are commonly used in the field, and the results are thus comparable to the results of previous experiments, most notably Artetxe et al. (2018b). The concatenated monolingual data consists of approximately 90 million sentences both in English and in German, and 1.82 billion words in English and 1.37 billion words in German. English and German are chosen as a language pair for a similar reason as the datasets: the pair is widely used as a benchmark in machine translation, and thus enables a fair comparison between the approaches. Sentences in the used newswire data are shuffled, and each sentence is on its own line. The training data consists of rather long sentences with an average length of approximately 30 words (Koehn and Knowles, 2017). The translation systems are evaluated using the *newstest2014* dataset. For the word embedding model, the *newstest2014* data is tokenized with Moses tools similarly as in the training set data. For the sense-aware model, I tokenize and word sense disambiguate the test data similarly to as in training the sense embedding-based model. From the results of the sense-aware translation model, I remove the sense annotations so that I have two data files for each sense-aware translation result, one with sense annotations and one without and calculate the BLEU score w.r.t. a reference file that is tokenized with the same script but not sense annotated. Therefore I am able to acquire the sense the model uses in translating the word but also get a more comparable BLEU score without the sense annotations in the translations. I also report BLEU scores of annotated files to find out whether annotations make a difference in the obtained BLEU scores.

4.1.1 Preprocessing

The first step in my pipeline is to train sense embeddings for both languages. The sense embedding models are then used as a sense inventory for word sense disambiguation of the data that is used as an input to the translation model. As noted, AdaGram is used to create the sense embedding models. AdaGram takes an input of two files: a tokenized data file, and a dictionary file. Because the News Crawl data is shuffled and I use the sense embedding model as a word sense inventory in the annotation process, tokenization is also performed line by line to maintain sentence boundaries.

The datasets are tokenized using standard Unix tools. The sentences are lowercased, different punctuation are removed, different space characters are normalized, and non-alphanumerics are deleted. Lastly, consecutive space characters are squeezed into a single space character. Tokenization is performed with a shell script that consists of the following Unix commands:

```
tr "[:upper:]" "[:lower:]" < $1 |
tr -d "[:punct:]" |
tr "\040\011\015" " " |
tr -cd "[:alnum:] \012" |
tr -s " " > $2
```

Non-breaking space characters were difficult to recognize beforehand, but were identified when reviewing the results of the preprocessing script. As the non-breaking space characters were encoded into the text as *nbsp* strings, the tokenized data included such anomalies as: *1800 average number of veteransnbspwho dienbspeach day* and *pilot ace computernbsp 8*. As seen from the examples, the non-breaking space characters encoded as text create unwanted noise in the data. The output includes 7 232 non-breaking space characters in the English data and 1 429 in the German data calculated with the following Unix command:

```
grep -o 'nbsp' filename | wc -l
```

Nbsp-strings seem to quite often occur consecutively as in the following example: *5 r kubica pol renault nbspnbspnbspnbspnbspnbspnbsp14015863*, and sometimes, although more infrequently, they form their own tokens or own lines. Because of training the sense embeddings models being rather expensive computationally, and because of the relatively low number of the

nbsp-strings in the data and their described nature of occurrences, I decided not to remove them.

An example of a "clean" sentence from the data looks like following: *iran isnt making an atomic bomb not at all chavez said monday.* As is seen in the example, all tokens are in lowercase, and contractions typical to English are tokenized so that the punctuation is removed, not changed into a space character which could be another solution. As a result, tokens including contractions like *isn't*, *wouldn't* or *I'll* appear in the data as *isnt*, *wouldnt*, and *ill*. In some cases, as in the last example, this could create a chance of misinterpretation for the model in the data, but in general, I wanted to have for example negations as one token and not include artificial tokens like subsequent *isn* and *t* in the data.

The German data is also tokenized with the same procedure. As a result, German data loses the diacritical marks, or the umlauts, as well as its eszett character, ß. Unfortunately, this was noticed only after translation was performed, and because of the time limits of the thesis, there was no time to train another model. However, the biggest effect of the tokenization scheme is expected to be on the BLEU score and the word-level review of the translation is still possible.

After preprocessing, dictionary files are made for both languages using the dictionary script provided with the AdaGram package[2] by Bartunov et al. (2016). The provided dictionary script creates a word frequency file with each word on their own line followed by a space and the word's frequency count in the data. The resulting input files consist of approximately 3.7 million unique tokens in English and 9.3 million unique tokens in German. In learning the sense embedding models for the languages with AdaGram, vocabulary is limited to tokens with more than 20 occurrences to remove noise generated by the low frequency words in the data which will lower the number of unique tokens.

4.2 Learning sense embeddings and sense annotating the data

Embedding models for different senses of each occurring word are learnt using AdaGram (Bartunov et al., 2016). The considered half-context size is set to 5. Dimensionality of the embedding vectors is set to 300. The maximum number of learned prototypes, or senses, is set to 5, and the minimum word frequency

[2]https://github.com/sbos/AdaGram.jl

is set to 20. Using 6 workers, the learning process for each language model takes approximately 72 hours in the CSC's Puhti computing environment.

To utilise the sense embeddings in the machine translation pipeline, I sense annotate the translation data using a script[3] that utilises the sense embedding models. This sense annotation process means that I perform a word sense disambiguation task on the data which I use as an input to the translation model. I iterate through the text data with the same window size that was used in learning the sense embeddings with AdaGram. For each word in the input data treated as the target word, I use the disambiguate-function included in the AdaGram library to disambiguate the word using the sense embedding models. I used a half-context size of 5 in learning the sense embeddings. Accordingly, I consider a maximum of 5 preceding and 5 posterior words as the context of the target word in the disambiguation task. The result of the disambiguate-function is an array of probabilities where each value is a probability of the prototype being the correct sense for the given word given the context. From the array of probabilities, I take the maximum value, and annotate the current target word with the index of the maximum value. Last, I write the annotated sentences into a new file. I perform the same procedure to both languages, and use the annotated sentences as monolingual inputs for the translation system.

After the annotation process, a sentence in the data looks like following: *iran2 isnt4 making4 an1 atomic2 bomb2 not3 at2 all2 chavez5 said3 monday5*. Each token is followed by an integer that identifies the annotated sense from the list of five possible senses. In the given example, *iran*, *atomic*, *bomb*, *at* and *all* are disambiguated as carrying the second sense of the words, *isnt* and *making* carry the fourth sense of the words, *an* is annotated with the first sense of the word, *not* and *said* are annotated as carrying the third sense of the words, and *chavez* and *monday* are annotated as carrying the fifth sense of the words. The concatenation of the integer produces potential problems with numeric tokens such as years or dates, since the numeric tokens are directly followed by an index identifier. The evaluation is performed on word-level translation of potentially ambiguous words selectively sampled by the author, so the problem with numeric tokens can be ignored in this work.

4.3 Monoses baseline

As presented in section 2.3, multiple efforts have been made in unsupervised statistical and neural machine translation. The statistical model of Artetxe

[3] `https://github.com/teemuvh/ma-thesis-scripts`

et al. (2018b) succeeded in exceeding the results of the neural model of Artetxe et al. (2018c) considerably as well as narrowing the gap between their supervised counterparts. It represents a successful model in unsupervised machine translation paradigm, and as such is an important baseline to compare to. In order to having a fair baseline to compare to, I train a statistical machine translation model using Monoses and aim at replicating the model and results presented in Artetxe et al. (2018b).

Two models are trained with Monoses, one with typical word embeddings and one with sense embeddings. Both models are trained with the default settings of the model presented in Artetxe et al. (2018b). As a part of Monoses pipeline, the input data is preprocessed and truecased using the Moses tools. Sentences with less than 3 or more than 80 tokens are removed. Embedding dimension is set to 300, and the Skip-gram context window is set to 5. MERT tuning is iterated for 10 epochs, and the number of back-translation iterations is set to 3. Training of the Monoses model takes approximately five days on the CSC Puhti computing environment.

4.4 NMT hybridization

My NMT model is built on top of the Monoses model so that I save the back-translations of the last iteration of back-translation process from the Monoses system, and utilise them as a synthetic parallel data for NMT. Similarly to Sennrich et al. (2016a), the back-translations are used on the source side while the target side consists of the parallel sentences from the original training data. The number of training sentences for the NMT model is 2 million as this is the parameter set for back-translation in Monoses. I use *newstest2008* as the validation data for NMT training to not have any overlap between training or test data and the validation data. The NMT model is based on Bahdanau et al. (2015a) and is implemented using OpenNMT. A more sophisticated model, for example a Transformer model (Vaswani et al., 2017) could as well be used but as the aim of the NMT hybridization is only to get a baseline with sense embedding method, it is not necessary in the scope of this thesis. In the NMT training, I use a bidirectional recurrent neural network (RNN) encoder-decoder model with bidirectional RNN on the encoder side and RNN on the decoder side. I use LSTM units with 0.3 dropout and two layers both on encoder and decoder side and stochastic gradient descent (SGD) as an optimizer. Hidden layer dimension is 1000, and word embedding dimension is 620. As an attention mechanism, I use MLP as in Bahdanau et al. (2015a) with 0.1 dropout. In this thesis, I do not

utilise byte pair encoding. The model is trained for 100 000 epochs.

As the BLEU score of the original SMT model is so low, I am not expecting to get a lot higher score with the NMT hybridization, but the test is performed to find out whether the SMT initialization combined with the NMT training yields better results in word-level translation which is evaluated qualitatively in this work.

5 Results & Discussion

In this section, I present the results of the sense embedding training as well as discuss the results of the sense embedding model. More importantly, I present the results of the translation models. The translation models are evaluated with BLEU, and the results are discussed. I also perform qualitative evaluation of the translation results to get a deeper insight on how the models perform in translating ambiguous words. The analysis of the results is included in the following subsections. First, in section 5.1, I present the results of the sense embedding model as well as discuss the results. In section 5.2, I present and analyse the results of the translation models.

5.1 Sense embeddings

To gain an intuitive estimation of how the models represent senses, I conduct nearest neighbour searches for a number of examples from the data. AdaGram package includes a function to calculate the cosine similarities of a set number of representations after which it outputs the results indicating the words that are semantically closest to the given target word. As I am operating with sense embeddings, I choose a number of words both in English and German, and calculate the nearest neighbours of the different senses AdaGram was able to induce from the data to see whether the inspected senses are meaningful. The English words evaluated within this section are *bank*, *atomic*, and *it*. The German word chosen for evaluation is *schlange*.

Neighbour	Sense	Closeness score
hsbc	0	0.7918...
citibank	0	0.7893...
wachovia	0	0.7871...
ubs	0	0.7842...
merrill	2	0.7833...
barclays	2	0.7830...
deutsche	0	0.7690...
dbkgnde	0	0.7684...
bacn	0	0.7624...
america	1	0.7543...

Table 2: Ten nearest neighbours of the first sense of *bank*.

The sense embedding model includes five different representations for the English word *bank*. Using the nearest neighbour search, I investigate how the model has clustered the different senses. The search retrieves a list of tuples, where each tuple contains the neighbouring word, its sense identifier, and its 'closeness' to the target word. Ten nearest neighbours of the first sense of the word *bank* are presented in table 2. Looking at the nearest neighbours, it seems clear that the induced sense has the meaning of a commercial bank.

Neighbour	Sense	Closeness score
banks	3	0.8386...
central	1	0.7627...
boe	1	0.7489...
ecb	1	0.7214...
governor	1	0.7166...
policymakers	0	0.7140...
policymaker	0	0.6950...
policy	1	0.6923...
rbi	1	0.6785...
reserve	4	0.6671...

Table 3: Ten nearest neighbours of the second sense of *bank*.

The nearest neighbours of the second sense of the word *bank* are presented in table 3. I interpret the second sense of the word *bank* to represent the sense of central bank as opposed to a commercial one represented by the first sense. The abbreviations boe, ecb and rbi possibly referring to the Bank of England, the European Central Bank, and the Reserve Bank of India, and the word *central* clearly reinforces the interpretation.

The nearest neighbours of the third sense of the word *bank* are presented in the table 4. The nearest neighbours clearly refer to the geographical meaning of the word, showing that the model can learn semantically unrelated meanings of the lexeme. The nearest neighbours of the third sense also show that the sense that was disambiguated in the sentence *four4 palestinians4 in3 the4 west4 bank3* appears to be the correct one given the context.

The fourth sense of the word *bank* has the nearest neighbours that are presented in table 5. The represented sense is not as clear as the first three. From the words picturing the actions or features of a bank, *accounts*, *account*, *deposits*, *banking*, and the word *institution*, the fourth sense can be interpreted to represent the meaning of the financial institution, not necessarily repres-

Neighbour	Sense	Closeness score
occupied	0	0.7878...
ramallah	1	0.7621...
hebron	0	0.7609...
nablus	0	0.7528...
jerusalemareas	0	0.7390...
jenin	0	0.7329...
jerusalem	0	0.7292...
settlements	0	0.7177...
warwon	0	0.7087...
israelioccupied	0	0.7047...

Table 4: Ten nearest neighbours of the third sense of *bank*.

Neighbour	Sense	Closeness score
hsbc	4	0.6589...
banks	0	0.6451...
accounts	1	0.6174...
account	1	0.6173...
santander	2	0.6162...
deposits	3	0.6088...
banking	4	0.6074...
rbs	3	0.6011...
barclays	3	0.5941...
institution	1	0.5911...

Table 5: Ten nearest neighbours of the fourth sense of *bank*.

enting the commercial banks as in the first sense, but the institution more generally.

The nearest neighbours of the last sense, as the maximum number of senses to learn is set to five, of the word *bank* are presented in table 6. The fifth identified sense for the word *bank* is already even more difficult to interpret. The nearest neighbours include abbreviations of commercial banks such as *icbc* and *wbk* but also words referring to the typical actions concerning the commercial meaning of a bank, such as *lender* or *investment*. Thus, the meaning gets a little overlapping with the first and the fourth ones and shows that learning too many representations can make the model harder to

Neighbour	Sense	Closeness score
lender	0	0.6766...
icbc	0	0.6370...
banks	4	0.6338...
zachodni	0	0.6005...
burdale	0	0.5959...
subsidiary	2	0.5957...
wbk	0	0.5785...
601939ss	0	0.5761...
investment	4	0.5715...
statebacked	0	0.5707...

Table 6: Ten nearest neighbours of the fifth sense of *bank*.

interpret. However, considering the used data is only from the newswire, the overrepresentation of the financial institution sense is understandable, and including data from other genres could result in more distinctive senses of the word *bank* identified.

Searching for the actual senses in a context shows the sentences where the senses occur. The first occurrence of the first sense of the word *bank* appears in the context window *energy1 nilesh1 shah4 of5 icici1 bank1 and1 sanjay2 nayar1 ceo2 of5*. I interpreted the first sense to carry the meaning of commercial bank. The occurrence of *ceo* of something could provide enough information about the commercial sense of the word. However, more context would be needed to make the interpretation. The second occurrence of the sense is in the context window *coming2 in5 aftermath1 of2 lasalle1 bank1 buy1*, which provides a clearer indication of the commercial sense. The second sense of the word *bank* appears first time in the context *have5 got4 to1 make5 the3 bank2 of3 england2 more3 cautious1*, strengthening the interpretation of the central bank meaning. The third sense appears first in the context *hamas1 loyalists1 in3 the4 west4 bank3 which4 is4 ruled3 by5 rival3*, which is a clear indication that the sense has been identified correctly, and the nearest neighbours are appropriate. The fourth sense of the word *bank* first appears in the context *by1 the2 queue1 for4 a3 bank4 of2 scotland1 cash4 dispensing1 machine4*, and the fifth sense appears in the context *provident2 bank5 also2 offers3 related2 financial3 services1*. As noted, the last two senses are already harder to interpret, and the context does not give much information about why it has been clustered into its own sense. In the end, for the purposes of machine translation, I expect the most meaningful property to be capability

Neighbour	Sense	Closeness score
iaea	0	0.8617...
viennabased	0	0.8280...
watchdog	0	0.7617...
amano	0	0.6977...
uns	2	0.6885...
yukiya	0	0.6564...
agency	1	0.6415...
energy	1	0.6284...
nuclear	3	0.6260...
abbasidavani	0	0.6234...

Table 7: Nearest neighbours of the first sense of *atomic*

to recognize the difference between the geographical and the financial institution, be it commercial or not, sense, and that is what the model clearly does.

Even though the maximum number of potential prototypes is set to five, stick-breaking process takes care of learning adaptively, meaning the number of learnt senses can be lower. For the word *bank*, the model learnt five senses, even though some of them might not have been very distinct. For the word *atomic*, the model only learns four different senses. Searching for the induced senses for the word *atomic*, the model returns the following values: 0, 0.382373; 1, 0.320396; 2, 0.135998; and 3, 0.161226, where the first number is the sense identifier and the second the probability truncated to the sixth decimal. Thus, the first sense, sense 0, has a prior probability of approximately 0.382373. Nearest neighbours of different senses of the word *atomic* represent four distinct senses of the word.

Ten nearest neighbours of the first sense of the word *atomic* retrieved from the model are presented in table 7. Many of the neighbouring words, such as *energy* and *nuclear*, are expected with the given target word, and do not tell much about the cluster itself. However, some words are more context dependent and reveal the underlying logic of the cluster. *Iaea*, the abbreviation of the International Atomic Energy Agency, *yukiya* and *amano*, the name of the former Director General of the IAEA, *watchdog*, and *abbasidavani*, the former head of the Atomic Energy Organization of Iran clearly indicate the first sense referring to what I name the political or the supervision context of *atomic*.

Neighbour	Sense	Closeness score
nuclear	0	0.8501...
weapons	3	0.8300...
tehran	0	0.8289...
program	0	0.7900...
nuclearweapons	0	0.7700...
nuclear	2	0.7530...
capability	1	0.7440...
uraniumenrichment	0	0.7408...
enrichment	0	0.7387...
iran	1	0.7361...

Table 8: Nearest neighbours of the second sense of *atomic*

Neighbour	Sense	Closeness score
hiroshima	0	0.7564...
nagasaki	3	0.7182...
atom	1	0.5943...
nuclear	1	0.5611...
bomb	1	0.5600...
plant	0	0.5514...
chernobyl	0	0.5282...
fukushimadaiichi	0	0.5255...
reactors	0	0.5235...
japans	1	0.5212...

Table 9: Nearest neighbours of the third sense of *atomic*

The second sense's ten nearest neighbours are presented in table 8. Clearly, the discovered sense is referring to the military sense of the word *atomic*, including such words as *weapons* and *nuclearweapons*.

The nearest neighbours of the third sense of the word *atomic* are presented in table 9. The words in the list indicate that the third discovered meaning cluster of the word *atomic* consists of a disaster meaning of the word whether being military or an accident as the list includes the cities of *Hiroshima* and *Nagasaki* as well as *Chernobyl* and the nuclear plant located the Fukushima prefecture in Japan: *fukushimadaiichi*.

Neighbour	Sense	Closeness score
atom	1	0.6472...
atoms	0	0.6232...
neutrons	0	0.6105...
fusion	3	0.6019...
electron	0	0.5987...
fission	0	0.5874...
photons	0	0.5831...
nanoscale	0	0.5807...
antiproton	0	0.5758...
ytterbium	0	0.5756...

Table 10: Nearest neighbours of the fourth sense of *atomic*

The last discovered sense of the word *atomic* has the nearest neighbours presented in table 10. The fourth sense cluster clearly represents the scientific meaning of the word *atomic*.

The first occurrence of the first sense that I named the political or the supervision is in sentence: *iran2 hoping2 to3 ward2 off5 any5 further4 sanctions5 on3 its1 oildependent1 economy4 agreed2 with2 the4 un5 international5 atomic1 energy2 agency2 iaea1 in3 august5 to3 clear2 up3 suspicions1 about4 its1 past5 secret3 nuclear3 activities2.* Again, the sense was actually learnt from the window of *agreed2 with2 the4 un5 international5 atomic1 energy2 agency2 iaea1 in3 august5.* Here, given the abbreviation *iaea* is present, I interpret that the meaning of *atomic* is pulled towards the supervision. The first occurrence of the second sense of the word *atomic* is in the sentence *iran2 isnt4 making4 an1 atomic2 bomb2 not3 at2 all2 chavez5 said3 monday5.* The sentence has the military meaning, and the word *atomic* is annotated with the sense identifier 2, meaning the model has identified the word to carry the second meaning. As the second meaning of the word was interpreted to be military, the word seems to be correctly disambiguated in this context. Interestingly, the meaning was identified as the military one even though the context included the word *bomb* which actually appears in the nearest neighbours of what I name the disaster sense. The third meaning of *atomic*, i.e. the disaster one, occurs first time in the context *tarnished1 by5 the4 sale5 of4 atomic3 secrets1 on1 a3 global5 black3 market2.* Here, the disaster sense is not easily interpreted by a human reader as the meaning could as well be for example the military one. The first occurrence of the fourth discovered meaning, which I interpreted as the scientific one, is in the

sentence *models2 offer4 solar1 power4 and2 atomic4 timekeeping1*. Probably the occurrence of the words *solar* and *power* lead the model to cluster the target word into the scientific sense. Looking at the examples given in this section, it is clear that the model has been able to differentiate distinct senses of the words *bank* and *atomic*. However, for some words, the model learns multiple senses in a situation where multiple senses would not necessarily be needed. For the lexeme *it*, the model learns five senses. The nearest neighbours of the senses show that there could potentially be two different senses if not only one. For this example, I only list the nearest neighbours without their sense identifiers and their probabilities, because the meaning of this example is only to show that the learnt senses are not always as well identifiable as in the previous examples.

The first representation has the following nearest neighbours: *company*, *its*, *announced*, *plans*, *gm*, *deal*, *sell*, *also*, *planned*, and *crr*. In comparison to the other induced senses, the first sense one only included one other pronoun in the nearest neighbours, when the others had multiple. The second sense has the following nearest neighbours: *really*, *this*, *just*, *i*, *actually*, *he*, *anyway*, *everything*, *quite*, and *something*. The nearest neighbours of the third sense are: *itself*, *literally*, *that*, *she*, *actually*, *however*, *he*, *everyone*, *its*, and *quite*. For the fourth one, the nearest neighbours are: *he*, *however*, *she*, *move*, *still*, *indeed*, *the*, *nonetheless*, *would*, and *although*, and for the fifth sense: *itself*, *however*, *this*, *actually*, *practicable*, *roadway*, *only*, *always*, *its*, and *desirable*. From these words, the meanings can not be distinguished similarly as with the words *bank* and *atomic*. It is clear that nouns have more meanings than the closed classes such as pronouns. Also adjectives can have multiple distinct meanings, WordNet lists three different meanings of the word *atomic*. The problem with the model thus lies in the fact that it is able to learn multiple representations for each word in the data even though it might not be needed. The number of learnt prototypes is dependent on the parameter α of the stick-breaking construction, and can be controlled in the AdaGram training. The value of α is set to the optimal of 0.1 evaluated in Bartunov et al. (2016). Setting the value lower than 0.1 could result in less prototypes per word (Bartunov et al., 2016), but the same effect would occur throughout the model, resulting in potentially too few representations in actually ambiguous cases. However, in the scope of this thesis, the fact that the model potentially learns too many prototypes for words that might not be as ambiguous does not matter, because the qualitative evaluation of the translation model is based on word-level translations of certain selectively sampled polysemous words. Even if machine translation sometimes fails for example in correctly translating pronouns, they rarely carry multiple meanings and the problem

could probably not be solved by the sense embedding method evaluated in this thesis.

The German example is the word *schlange*, which could carry the meaning of a queue or a snake. By searching for the word sense probabilities in the German word embedding model, it seems that the model has learnt two distinct senses for the word, one with a probability of approximately 62.55 and one with approximately 37.45. The nearest neighbours of the first distinguished sense are *warteschlange*, *schlangen*, *menschenschlange*, *menschenschlangen*, *warteschlangen*, *geduldig*, *tr*, *einlass*, *wartender* and *kassenhuschen*. As the German data lost the umlauts in the preprocessing procedure, the words do not include them in the model. Thus, *tr* is probably *tür* and *kassenhuschen* is most probably *kassenhäuschen*. From the nearest neighbours, it is easy to interpret that the meaning of the first sense is the one of queue. The nearest neighbours of the second sense are *katze*, *ratte*, *schildkrte*, *spinne*, *fledermaus*, *giraffe*, *raubkatze*, *eule*, *ziege* and *echse*. Again, *schildkrte* is most probably *schildkröte*. All the nearest neighbours of the second sense are animals so it seems safe to assume that the second sense carries the meaning of a snake.

In this section, I gave some examples and shortly presented the results of the sense embedding models, showing some of their strengths and weaknesses. The analysis shows that the models are able to learn distinct meanings for polysemous words, which can be expected to help in translating ambiguous words. In the next section, I analyse the results of the translation models and investigate whether my hypothesis holds.

5.2 Translation

As stated in chapter 1, my hypothesis is that the word level translations of ambiguous words can be more adequate when sense embeddings are used in mapping the embeddings into a shared vector space than when the mapping is conducted with single-sense word embeddings. In this section, I present and analyse the results of the different translation models I trained during this thesis. In regards to typical automatic machine translation evaluation metrics, I do not expect significant increases. Nevertheless, I use automatic evaluation for comparison between the previous model that the model in this thesis bases on, namely Artetxe et al. (2018b), and my proposed models. I also qualitatively analyse a set of randomly sampled translations to get some general insight about the overall quality of the translations. In addition, I analyse a selectively sampled set of example translations of sentences that include potentially polysemous words which I expect to be difficult to

	DE-EN	EN-DE
Monoses (Artetxe et al., 2018b)	17.43	14.08
Baseline	14.88	10.95
SenseSMT	10.98	6.29
BTNMT	9.38	5.24

Table 11: BLEU scores of the models

translate by a typical word sense-based translation model. The investigated polysemous target words are found from Rios et al. (2018). The objective of the qualitative analysis is to gain insight on whether the models produced different solutions in translating the polysemous target words in the selected sentences, and whether the sense embedding-based model created more adequate translations. First, I present the BLEU scores of the models trained during this thesis, after which I continue to the qualitative analysis.

The automatic comparison of the models is performed in lowercased tokenized BLEU score, similarly to in Artetxe et al. (2018b), even though the tokenized BLEU is not necessarily recommended since the results can differ based on the used tokenization script. BLEU scores are obtained with Moses package's *multi-bleu.perl* and the results are presented in table 11. I include four models in the results table: the Monoses model presented in Artetxe et al. (2018b), my baseline model that is an attempt to recreate the original Monoses model, and the two proposed sense embedding-based models: the SMT model and the NMT model trained with the back-translated data. BLEU scores of the sense embedding-based models are calculated using the translated files with sense annotations removed w.r.t. their reference files tokenized with the same script as the original sense embedding training data.

The result of my recreation of the original Monoses model yields a BLEU score of 14.88 and 10.95 in German-English and English-German directions respectively. The scores are approximately 2.55 points lower in German-English direction and 3.13 points lower in the English-German direction than what Artetxe et al. (2018b) report. The models are trained with the same concatenated WMT's translation task's News Crawl 07–13 data and tested with the same *newstest2014* data using the lowercased *multi-bleu.perl* script. Both models, Artetxe et al. (2018b) and mine, are trained with the same hyperparameters and unsupervised tuning and back-translation iterations are similar. It is difficult to say what creates such a high gap between the original model and mine. Clark et al. (2011) point out the instability of optimizers,

especially MERT, used in translation but a gap of nearly 3 points or more is still quite huge even when optimizer instability is considered. Another run of 10 iterations of MERT tuning and 3 iterations of back-translation obtains a BLEU score of 14.96 and 10.71 in German-English and English-German respectively. The German-English direction increased a little while the English-German direction decreased. However, the change is so small that according to this experiment, the drop between the scores reported in Artetxe et al. (2018b) and my baseline model are not explained by optimizer instability but I will accept the results as they are. Investigated in Wieling et al. (2018), reproducibility is an issue in NLP. Wieling et al. (2018) present that even if the source code and the used data is available, like in the case of this work, it is not certain that the results can be successfully reproduced.

The SMT model trained with sense embeddings achieves a BLEU score of 10.98 in German-English direction and 6.29 in the more difficult English-German direction when the sense annotations are removed from the translated sentences and compared to the reference data tokenized the same way as the sense-aware model's training data. The negative BLEU scores could be explained by the different tokenization method used in training the sense-aware model compared to training the word-embedding based model. The baseline Monoses model was trained with data that was tokenized using Moses tokenizer developed for each language used in training while the sense embedding-based model was trained with data tokenized with a more general tokenization script that resulted in a somewhat ill-formed German data. Further, the difference in BLEU score of the German-English direction can occur because of a different way of handling tokens. Moses tokenizer splits words at punctuation resulting in such forms as *can* and *t* in a negation of the verb *can* while I decided to merely remove punctuation in sense embedding training resulting in *cant* using the same example. Thus, consistent tokenization in both schemes, the word embedding-based model and the sense embedding-based one, would possibly pull the BLEU scores closer to each others. The decrease in BLEU scores is rather close between the German-English and English-German directions when the baseline and the sense-aware models are compared, the drop being 3.90 and 4.66 respectively. I expect that using language specific Moses tokenizer in sense embedding training could yield better results in general in addition to a lower drop between the models but unfortunately there was no time for another run of the complete pipeline due to time constraints of this work. Also, the used word sense disambiguation script removes the out-of-vocabulary (OOV) tokens, the tokens that do not appear in the sense embedding model, from the data. This can result in an unnecessary mismatch between the data in situations where some tokens

for example occur in the English data but their correspondent words do not occur at all or do not occur frequently enough in the German data. Perhaps, instead of removing, changing them to an artificial OOV token such as *unk* could possibly result in a better score.

The NMT model trained with the back-translated data from the SMT model yields the lowest score of all the models acquiring 9.38 and 5.24 BLEU score in German-English and English-German directions respectively. The lower scores do not come as a surprise considering the relatively low score of the SMT model that initializes the sentences used as a parallel data in training the NMT model. Perhaps annotating tokens with sense identifiers has increased the number of rare words so high that the NMT model struggles in learning and as a result the translations include a lot of unknown tokens which affects the obtained BLEU score. Negative results of the NMT training are quite disappointing since significant increase in BLEU scores has been noticed between an SMT and a hybridization model before (for example Artetxe et al. (2019)). I suggest using a larger set of back-translated sentences, the number now being 2 million sentences in each language, for better results as NMT is known to demand large sets of data for training. Also, utilising byte pair encoding in the NMT training could possibly yield better BLEU scores.

The results of the sense embedding-based models decrease even more when the models are evaluated with sense annotations. For this, the test data is sense annotated and the translations are evaluated w.r.t. the sense annotated data. Here, the BLEU scores of the sense embedding-based SMT model are only 4.96 and 3.33 in German-English and English-German directions. The NMT model scores even lower with BLEU scores of 3.97 and 2.67 in German-English and English-German directions respectively. Table 12 shows a lowercased example of an annotated sentence from the reference data with its translations from the annotated translation data. From this example, it can be noticed that the annotations have translated rather well into English as four out of nine tokens have the exact same translation between the sense-aware SMT model and the reference data, while only one of the matching tokens has a different sense annotation. Quite similarly, five words have the same translation in the NMT model's result as in the reference data. However, the NMT model's result has one more token than the reference, and one of the matching tokens has a different sense annotation. The token that is differently annotated is the same in both translations: *like*. In both translations it has the sense identifier 2 while in the reference sentence the sense identifier is 4. In the sense embedding model, *like* has five induced senses. The fourth sense that the token in the reference translation was

disambiguated into has the following ten nearest neighbours: *little*, *vaguely*, *actually*, *tiny*, *something*, *looks*, *strange*, *resembles*, *makes*, *feels*, where words like *looks* and *resembles* are somewhat semantically close to the meaning the reference sentence carries. Ten nearest neighbours of the second sense are *other*, *including*, *such*, *unlike*, *as*, *include*, *example*, *include*, *socalled* and *both*. While the German word in the original sentence is *wie*, I interpret the second sense to be closer to the original source word based on the nearest neighbours. In this example the senses are not obvious. The sense disambiguated in the reference sentence for the word *movie* has the nearest neighbours of *film*, *movies*, *films*, *cinema*, *hollywood*, *sequel*, *documentary*, *horror*, *blockbuster* and *hobbit*. The SMT model translated the German word *film* as *film* with the sense identifier 4. The nearest neighbours of the sense are *movie*, *films*, *documentary*, *movie*, *movies*, *horror*, *filmmaker*, *genre*, *cinema* and *comedy*. From the nearest neighbours it seems like the used word carries the very same meaning in both translations. However, three of the induced senses of the English word *film* are quite overlapping so any of senses 3–5 would have been suitable. Sense 1 carries an obvious meaning of film festivals with such nearest neighbours as *festival*, *cannes*, *toronto* and *sundance*, while sense 2 has a lot of movie genres in the nearest neighbours. Also sense number 4 of the word *movie* has a lot of genres in the nearest neighbours, while the rest of the senses are quite similar and overlapping. Thus, the annotations have translated rather well into the sense embedding-based result and the NMT one. The source sentence is *die2 szene3 sei1 wie2 in1 einem2 film3 gewesen2*. The German word *wie* has the sense identifier 2 in the disambiguated data, and that could be the reason the translations also have the identifier 2 attached. The word *film* has three induced senses in the German model. However, they seem to be overlapping containing some same words in the nearest neighbours and it is not easy to figure out the difference in the meaning clusters. One observation can still be made: both, the sense 3 of the German word *film* and the sense 4 of the translated word *film* have *komdie* and *comedy* in the nearest neighbours so it can be proposed that the translation model has used the correct sense although the evidence is not very convincing. The previous examples are not very fruitful in comparing the word-level translation of ambiguous words since the meaning clusters are not distinguishable enough for a proper analysis. The example was only to show how the annotations may affect the obtained BLEU scores which of course decrease if the annotations differ in the reference and the translated sentences if the other factors are constant. Next, I review the translation results in more general and come back to the word-level translation later in this chapter.

reference	*the3 scene3 was4 like4 something4 out5 of1 a2 movie3*
SenseSMT	*the3 scene3 was4 like2 id1 been3 in2 a2 film4*
BTNMT	*the3 scene3 was4 like2 to4 have2 been3 in2 a2 movie3*

Table 12: Sense annotated translations

source	*Die Szene sei " wie in einem Film " gewesen .*
reference	*The scene was " like something out of a movie " .*
baseline	*The scene was " like being in a movie . "*
SenseSMT	*The3 scene3 was4 like2 id1 been3 in2 a2 film4*
BTNMT	*the3 scene3 was4 like2 to4 have2 been3 in2 a2 movie3*

Table 13: Translation table 1

Examples of source sentences with their reference and translated sentences are shown in tables 13–15. The source sentences and the reference sentences are presented in Moses tokenized form and the translations in the same form as they occur in the translation output. The translations are investigated in German-English direction. The models perform slightly better on the German-to-English direction, which should consequently result in better examples for the qualitative evaluation. The sampling has been conducted with a random number generator to get truly random examples.

Table 13 includes the source sentence *Die Szene sei " wie in einem Film " gewesen .* and its reference and contrastive translations. In general, the translation quality is quite good. The meaning of the sentence can easily be comprehended from any of the translations. The fact that each translation includes the word *like* shows that the translations maintain the metaphorical nature of the source sentence.

Table 14 shows another example of a rather good translation result. Interestingly, the reference sentence assigns the volume in a different unit, which consequently changes the numeral from 62.3 billion to 2.2 trillion. The baseline and the sense-aware SMT model get the numerals correctly but change the units into gallons and barrels respectively. The NMT model loses the amount and the object of the sentence by changing them to OOV tokens. The sentences still maintain a reasonable level of understandability throughout the translations albeit having some typical problems for machine translation.

Table 15 presents an example of a more difficult to translate sentence with

source	*Die Pipeline ist auf eine jährliche Kapazität von 62,3 Milliarden Kubikmetern Erdgas ausgelegt .*
reference	*The pipeline is designed for an annual capacity of 2.2 trillion cubic feet of natural gas .*
baseline	*The pipeline is a yearly capacity of 62,3 designed billion gallons of gas .*
SenseSMT	*The5 pipeline4 is4 to2 an5 annual5 capacity1 of2 6231 designed4 billion4 barrels4 of2 natural1 gas4*
BTNMT	*the5 pipeline4 is4 set2 to2 <unk> an5 annual5 capacity1 of2 <unk> billion4 tons2*

Table 14: Translation table 2

source	*An dem Projekt sind auch die Industrie- und Handelskammer in Neubrandenburg sowie der Deutsche Hotel- und Gaststättenverband (Dehoga) Mecklenburg-Vorpommern beteiligt .*
reference	*Also involved in the project are the Chamber of Industry and Commerce in Neubrandenburg and the German Hotel and Restaurant Association (Deutsche Hotel- und Gaststätte , Dehoga) of Mecklenburg-Western Pomerania .*
baseline	*Also on the project are the Chamber of Commerce in Anchorage , including the National Housing and Planning Association (Maine) involved .*
SenseSMT	*On5 a3 project2 that4 are3 the5 chamber1 of5 commerce1 and5 industry1 in5 central4 connecticut4 us5 hotel5 and5 restaurant3 association4 involved4 in3 colorado3*
BTNMT	*the5 initiative3 that4 are3 in5 the5 industry1 and2 commerce1 industry1 including3 the4 us5 hotel2 <unk> and2 <unk> owned4 by4 colorado3*

Table 15: Translation table 3

a lot of named entities that are typically quite difficult for MT systems. As is seen in the translations, the named entities get mixed: *Dehoga* turns into *Maine* and *Neubrandenburg* into *Anchorage* in the baseline model's translation, *Neubrandenburg* turns into *central connecticut* in the sense embedding-based SMT model and the milieu is *colorado* in the NMT model's translation. On a positive note, *the Chamber of Industry and Commerce* has been translated quite well in the baseline and the SMT models into *the Chamber of Commerce* and *the5 chamber1 of5 commerce1 and5 industry1*. The theme was somewhat lost in the baseline translation as the *Hotel and Restaurant Association* turned into *Housing and Planning Association*. SMT and NMT models seem to have maintained the theme even though the NMT model includes an OOV token as the other part of the conjunction. However, the translations presented in table 16 have failed in terms of fluency and ad-

equacy, and the original meaning is not conveyed to the translations. Based on the examples presented in tables 13–15, the translations are still rather good taken the low BLEU scores into consideration. In the following part of this chapter, I evaluate a set of selectively sampled sentences that include a potentially ambiguous word that I expect to be difficult to translate and see whether the sense embedding method has made any difference in translating the ambiguous words by investigating the translations as well as getting the nearest neighbours of the ambiguous words from the sense embedding models and reviewing the sense clusters the used senses belong to in the embeddings. The sentences are presented in the form they appear in the translated data. Thus, umlauts and eszett characters are missing in the German examples taken from the sense embedding-based models. The results are presented in tables 16–18.

source	*Ohnehin sei die Zubereitung veganer* ***Gerichte*** *weitaus günstiger und weniger zeitaufwendig , als gemeinhin angenommen .*
reference	*In any case , the preparation of vegan* ***dishes*** *is much cheaper and less time consuming than commonly assumed .*
baseline	*That was because the cooking veganer* ***courts*** *far cheaper and less time-consuming than generally thought .*
SenseSMT	*Otherwise3 the3 diet3 vegans1* ***dishes1*** *was5 far4 less1 expensive1 and4 more4 timeconsuming1 than4 commonly3 thought5*
BTNMT	*otherwise3 the3 <unk> <unk>* ***dishes1*** *was5 less1 expensive1 and4 more4 expensive1 than4 commonly3 thought5*

Table 16: Contrastive translations 1

Table 16 shows a set of contrastive translations between the German sentence and the reference and machine translations with the word in focus in bold. The focus word in the first example is *Gerichte*, which could be translated into the plural forms *courts* or *dishes* in English. The reference sentence has the word *Gerichte* translated as *dishes*. The baseline model however translates the word as *courts*, while the sense embedding-based models correctly translates the word as *dishes1*. The word *Gerichte* has three senses in the German sense embedding model. The first sense has the following nearest neighbours: *justiz*, *richter*, *staatsanwlte*, *strafverfolger*, *behrden*, *rechtsprechung*, *gerichte*, *urteile*, *gerichtsbarkeit* and *fachgerichte*. The first sense clearly carries the judiciary meaning of the word. The nearest neighbours of the second sense of the word in the German sense embedding model are *speisen*, *kstlichkeiten*, *desserts*, *salate*, *zubereitet*, *kche*, *suppen*, *beilagen*, *vegetarische* and *saucen*. Clearly the carried meaning of the sense is the culinary one. The third sense is a little overlapping with the first one with

the nearest neighbours *richter*, *urteilen*, *gerichte*, *landgerichte*, *zivilgerichte*, *arbeitsgerichte*, *verwaltungsgerichte*, *straburger*, *oberlandesgerichte* and *urteile*. Thus, to evaluate whether the sense embedding method made the difference in this example, the German word should be annotated with the sense identifier 2 in the annotated German test data so the token to translate would carry the culinary meaning. Searching for the sentence in the word sense disambiguated test set shows that the word *gerichte* has been annotated with the correct sense identifier during the WSD process as the sentence in the data is *ohnehin1 sei4 die2 zubereitung2 veganer1 gerichte2 weitaus1 gnstiger2 und1 weniger4 zeitaufwendig1 als4 gemeinhin1 angenommen1*. Thus, I believe the translation model has used the correct sense to identify that the translation is supposed to be *dishes* as opposed to *courts* that was used in the word embedding-based baseline model's translation.

Table 17 presents another example of contrastive translations with the focus word on bold. The German word *Kurs* could translate as *price* or as *course* or *class* in English. Here, the reference sentence has the word *course* as the translation. The word embedding-based model has translated the word as *price* while both of the sense embedding-based models have correctly used the word *course* in the translation. The word *Kurs* has three senses in the German sense embedding model. The nearest neighbours of the first sense are *regierungskurs*, *modernisierungskurs*, *politikstil*, *oppositionskurs*, *schlingerkurs*, *kursschwenk*, *eurokurs*, *atomkurs*, *europolitik* and *sparkurs*. I interpret the meaning to be something of a direction. The second sense has the following nearest neighbours: *aktienkurs*, *zeitweise*, *goldpreis*, *brsenkurs*, *commerzbankaktie*, *vwaktie*, *appleaktie*, *greenback*, *thyssenkruppaktie* and *celesioaktie*. The sense clearly carries the financial meaning of the word. The nearest neighbours of the third sense are *5419*, *stadtkurs*, *hochgeschwindigkeitskurs*, *rundkurs*, *5073*, *bergundtalkurs*, *segeltrn*, *skirollern*, *trn* and *innenstadtkurs*. I interpret the sense cluster to have the meaning of some sort of a track. The third sense is also the one used in the word sense disambiguated German test data. Thus, it is interesting the sense-aware models have translated the word correctly since the German model does not

source	*Der kombinierte* ***Kurs*** *aus englischer Literatur und Sprache wird abgeschafft .*
reference	*A combined English literature and language* ***course*** *will be scrapped .*
baseline	*The combined* ***price*** *from English literature and language is abolished .*
SenseSMT	*The3* ***course2*** *flowing1 from5 the2 english1 language3 and4 literature2 is2 abolished1*
BTNMT	*the3 <unk>* ***course2*** *from5 english1 literature2 and4 english1 will3 be2 abolished1*

Table 17: Contrastive translations 2

source	*Die **Wahl** wurde zu einem Erdrutschsieg und Nachrichtensprecher hielten inne , um die historische Bedeutung dieser Stunde vor Augen zu führen .*
reference	*The **election** turned out to be a landslide , and news anchors paused to reflect on the historic nature of the hour .*
baseline	*The **choice** was to a landslide and Nachrichtensprecher appointed , keeping the historical importance of this hour to bring to mind .*
SenseSMT	*The4 **election3** was5 to3 create4 a1 landslide1 over4 presenter3 and4 kept4 occupied4 the2 historical3 significance1 of3 this4 hour4 in2 mind2 to1 follow4*
BTNMT	*the4 **election3** was5 <unk> to3 a1 <unk> and4 leaving5 the2 historic3 meaning3 to3 lead2 this4 hour4 of3 view2*

Table 18: Contrastive translations 3

include the educational sense of the word. However, the translated sense of the word *course*, sense 2, does not carry the educational meaning of the word in the English sense embedding model either. The nearest neighbours of the second sense are *ninehole*, *layout*, *par72*, *courses*, *par71*, *par70*, *9hole*, *pinx*, *cordevalle* and *par73*, clearly carrying a golf course meaning of the word. In the word sense disambiguated English test data, the word is annotated with the sense identifier 4, which has the nearest neighbours: *courses*, *semester*, *semesters*, *intensive*, *16week*, *pgce*, *semesterlong*, *coursework*, *seminars* and *syllabus*. This is obviously the correct sense that should have been carried over to the translation. As was already noted in table 12, the annotations may introduce some troubles to the MT systems as they might be carried wrongly into the translations. Still, the example shows that the sense-aware model has translated the ambiguous word correctly as *course*, albeit with a wrong sense identifier, while the word embedding-based model has not.

Table 18 presents one more example of contrastive translations. The German word *Wahl* could translate into English as *election* or *choice*. In the reference translation the word *election* has been chosen as the translation. The baseline model translates the word as *choice* while the sense-aware models translate it as *election* with the sense identifier 3. The word *wahl* has four induced senses in the German sense embedding model. The first sense has the following nearest neighbours: *kommunalwahl*, *urnengang*, *landtagswahl*, *wahlen*, *bundestagwahl*, *bundestagswahl*, *nrwlandtagswahl*, *europawahl*, *prsidentenwahl*, *brgerschaftswahl*. The sense clearly carries the election meaning of the word. The second sense has the following nearest neighbours: *entscheidung*, *wahlen*, *bundestagswahl*, *prsidentenwahl*, *abstimmung*, *wahl*, *meinungsbildung*, *partei*, *richtungsentscheidung* and *auswahl*. The second sense could be interpreted to carry the meaning of decision or choice because of the neighbouring words such as *entscheidung* and *meinungsbildung*. The

nearest neighbours of the third sense are *nominierung*, *vereidigung*, *ernennung*, *kr*, *staatsprsidenten*, *ernennung*, *ernennung*, *neuen*, *vereidigung* and *prsidenten*, and the nearest neighbours of the fourth sense are *neuwahl*, *gewhlt*, *kr*, *wiederwahl*, *wahlen*, *miss*, *ersatzdelegierten*, *kandidaten*, *gekrt* and *kandidatenliste*. The last two senses are overlapping with the first one and I interpret them to also carry the election meaning of the word. Thus to see whether the sense embedding have potentially influenced the translation, I find the corresponding sentence from the sense annotated German test data to see which sense has been identified for the current target word during the WSD process. The sentence is word sense disambiguated as *die5 wahl1 wurde5 zu4 einem5 erdrutschsieg1 und1 nachrichtensprecher1 hielten1 inne1 um3 die1 historische2 bedeutung1 dieser3 stunde4 vor2 augen4 zu3 fhren2*. The induced sense for the word *wahl* is the first one. Thus, the translation to *election* is correct also in the light of the sense embedding model's WSD process and has been translated correctly by the sense embedding-based models.

The examples presented in tables 16–18 show that utilising sense embeddings instead of single-sense word embeddings can turn out to be crucial in translating ambiguous words. In all analysed examples, the ambiguous target word has been translated correctly by the sense-aware models while the word embedding-based model has failed in translating the word. Occurring problem however is that the sense identifiers get a little mixed up in the translation process, which results in lower BLEU score when calculated w.r.t. a word sense disambiguated test data. Still, when word-level translation of ambiguous words is considered, it seems that the sense embedding method can have a substantial effect on the quality of the translation results.

6 Conclusion

In this thesis, I have trained two monolingual sense embedding models in order to train sense-aware machine translation systems. Unlike typical word embedding-based translation models, the proposed sense-aware models utilise sense embeddings in mapping the monolingual word embeddings into a shared vector space in order to be able to train an unsupervised translation model. The aim was to answer to the following research question: does integrating a word sense disambiguation module into an unsupervised machine translation pipeline result in better translation quality of ambiguous words in comparison to a current state-of-the-art unsupervised machine translation system and thus increase the adequacy of the translation in general? The analysis performed in chapter 5 demonstrates that the sense embedding method can in fact result in better translation quality when the translations are qualitatively evaluated on the word-level. The sense-aware models succeed in translating polysemous words in German-English direction in the evaluated examples where the word embedding-based model fails. Some problems occur in translation of sense annotated forms because it seems the models sometimes mix up the annotations or the annotated forms during translation. The effect has been noted in the analysis chapter. Still, the models translate the ambiguous target words correctly but the results may, in some cases, have wrong sense identifiers. In conclusion, this thesis gives preliminary results indicating that the sense embedding method can be beneficial in translating ambiguous words in English-German language pair evaluated in German-English direction.

As the method is not limited to unsupervised MT, but can be incorporated to other paradigms as well, future research could include integrating a similar WSD module into a supervised MT system which could result in better translation results than the currently lower-scoring unsupervised models. Also, a language specific tokenization method should be included in the future research of the method to ensure better results. Better BLEU scores could be obtained in the NMT paradigm by utilising byte pair encoding, which has had notable results when used in machine translation. Last, the method can be investigated with different language pairs or more interestingly in different domains, such as literary translation, where sense-awareness could result in better translations in general.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1042. URL `https://www.aclweb.org/anthology/P17-1042`.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, 2018a.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, November 2018b. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*, April 2018c.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1019. URL `https://www.aclweb.org/anthology/P19-1019`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015a. URL `http://arxiv.org/abs/1409.0473`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015b.

Sergey Bartunov, Dmitry Kondrashkin, Anton Osokin, and Dmitry Vetrov. Breaking sticks and ambiguities with adaptive skip-gram. In *Artificial Intelligence and Statistics*, pages 130–138, 2016.

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. Neural versus phrase-based machine translation quality: a case study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1025. URL https://www.aclweb.org/anthology/D16-1025.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X.

Jose Camacho-Collados and Mohammad Taher Pilehvar. From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Int. Res.*, 63(1):743–788, September 2018. ISSN 1076-9757. doi: 10.1613/jair.1.11259. URL https://doi.org/10.1613/jair.1.11259.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014a. Association for Computational Linguistics. doi: 10.3115/v1/W14-4012. URL https://www.aclweb.org/anthology/W14-4012.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014b. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://www.aclweb.org/anthology/D14-1179.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-2031.

Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

Ignacio Iacobacci and Roberto Navigli. LSTMEmbed: Learning word and sense representations from a large semantically annotated corpus with long short-term memories. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1685–1695, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1165. URL https://www.aclweb.org/anthology/P19-1165.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. SensEmbed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 95–105, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1010. URL https://www.aclweb.org/anthology/P15-1010.

Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1176.

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

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3204. URL https://www.aclweb.org/anthology/W17-3204.

Philipp Koehn, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133, 2003. URL https://www.aclweb.org/anthology/N03-1017.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*, 2018a. URL https://openreview.net/forum?id=rkYTTf-AZ.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations*, 2018b.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018c.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 1284–1290. AAAI Press, 2015a. ISBN 9781577357384.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Topical word embeddings. 2015b.

Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, page 133–139, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118711. URL https://doi.org/10.3115/1118693.1118711.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation, 2013b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013c. Curran Associates Inc.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013d. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1090.

Andriy Mnih and Geoffrey E Hinton. A scalable hierarchical distributed language model. In *Advances in neural information processing systems*, pages 1081–1088, 2009.

Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1113. URL https://www.aclweb.org/anthology/D14-1113.

Dai Quoc Nguyen, Dat Quoc Nguyen, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. A mixture model for learning multi-sense word embeddings. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 121–127, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-1015. URL https://www.aclweb.org/anthology/S17-1015.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. Improved alignment models for statistical machine translation. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999. URL https://www.aclweb.org/anthology/W99-0604.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural*

Language Processing (EMNLP), pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Annette Rios, Mathias Müller, and Rico Sennrich. The word sense disambiguation test suite at WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 588–596, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6437. URL https://www.aclweb.org/anthology/W18-6437.

Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, pages 11–19, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4702. URL https://www.aclweb.org/anthology/W17-4702.

John I Saeed. *Semantics*. John Wiley & Sons, 2015.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016a. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://www.aclweb.org/anthology/P16-1009.

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 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://www.aclweb.org/anthology/P16-1162.

Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4(2):639–650, 1994. ISSN 10170405, 19968507.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231, 2006.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112.

Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Martijn Wieling, Josine Rawee, and Gertjan van Noord. Squib: Reproducibility in computational linguistics: Are we willing to share? *Computational Linguistics*, 44(4):641–649, December 2018. doi: 10.1162/coli_a_00330. URL https://www.aclweb.org/anthology/J18-4003.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Yadollah Yaghoobzadeh and Hinrich Schütze. Intrinsic subspace evaluation of word embedding representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 236–246, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1023. URL https://www.aclweb.org/anthology/P16-1023.