UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Andre Tättar

Unsupervised Machine Translation using Cross-Lingual N-gram Embeddings

Master's Thesis (30 ECTS)

Supervisor: Mark Fishel, PhD

Tartu 2018

Juhendamata masintõlge kasutades keeltevahelisi fraaside vektoresitusi

Lühikokkuvõte: Praegused parimad masintõlke süsteemid saavutavad suurepäraseid tulemusi, kuid nõuavad tulemuste saamiseks suuri paralleelkorpusi. Palju tööd on tehtud, et saada häid tõlketulemusi väiksese paralleelkorpusega keeltepaaridele, aga võrreldavaid tulemusi suurte paaralleelkorpusega keeltele pole saadud. Selles töös ma pakun välja uudse süsteemi, mis teeb juhendamata masintõlget kasutades n-grammide (fraaside) vektoresitusi, mille abil õpitakse keeltevahelised fraaside vektoresitused. Minu lahendus nõuab ainult ühekeelseid korpuseid. Ma raporteerin oma tulemused eesti - inglise - eesti keelepaari vahel. Arendatud süsteem ei tööta nii hästi kui loodetud, aga testide järgi võib öelda, et see töötab paremini kui sõna-sõnalt otse tõlkida.

Võtmesõnad: Juhendamata masintõlge, n-grammide vektoresitused, keeltevahelised n-grammide/fraaside vektoresitused

CERCS: P176, Tehisintellekt

Unsupervised Machine Translation using Cross-Lingual N-gram Embeddings

Abstract: The current best machine translation systems have achieved excellent results, but rely heavily on large parallel corpora. There have been many attempts on getting the same good results on low-resource languages, but these tries have been somewhat unsuccessful. In this work, I propose a novel unsupervised machine translation system that uses n-gram embeddings for getting the translations by learning cross-lingual embeddings. This solution requires only monolingual corpora, not a single parallel sentence is needed, which is achieved by using unsupervised word translation. I report my findings for Estonian - English - Estonian language pair. The solution does not work as well as expected, but tests suggest that it works better than simple word-by-word translation.

Keywords: Unsupervised machine translation, N-gram/phrase embeddings, Cross-lingual n-gram/phrase embeddings

CERCS: P176, Artificial Intelligence

Contents

1 Introduction

1.1 Objective

The solution for machine translation (MT) historically has been a supervised task, which means that parallel corpora are required for training. At first, MT was done by using phrase-based machines, then came statistical machine translation and for last 5 years or so, machine translation is done by neural networks - neural machine translation (NMT). NMT has very good performance, but there is a drawback - with every new way of doing MT, the necessity of parallel corpora has increased. The drawback here is that many language pairs have zero or minimal amount of parallel data, called low-resource language pairs. The main objective of this thesis dissertation is to develop a method for doing machine translation with low-resource language pairs. Furthermore, I intend to make a solution that is simple and open-source, so anyone willing could use it and train their own system. An open system could also spark the interest for collective improvement to achieve better results.

1.2 Motivation

The motivation for unsupervised MT comes firstly from Finno-Ugric languages - many languages are becoming extinct, like Inari Sami (300 speakers left) or are extinct, like Livonian (June 2, 2013). There is very little data about these languages and a translation system could keep the languages alive.
Secondly, we could say that Estonian-English is not a low-resource language pair, but any other language pair like Estonian-Latvian is a low-resource language pair and state-of-the-art machine translation for Estonian-Latvian use English as the pivot language, so Estonian gets translated into English and then into Latvian. In a more general view, there are many low-resource languages in the world and unsupervised MT could help all of them.
Thirdly, University of Tartu is helping with the organization of WMT (wmt) conference, which is one of the most important conferences for MT. WMT datasets are often used to compare results, most notably English-French language pair from WMT'14 and German-English language pair from WMT'16. I am aiming to compete with this work at the WMT'18 conference unsupervised translation task.

1.3 Contribution

In this thesis the main contribution can be resolved to:

- Machine translation using zero parallel sentences and no seed dictionary - in other words completely unsupervised machine translation. This is a novel and promising idea for solving the task of MT and interlingua in general.

Unsupervised MT makes use of monolingual data, which is easy to find and cheap. Cheap in the sense that parallel data requires human work to translate sentences, which is expensive. For example, English-Estonian language pair has about 10^6 of non-noisy parallel sentences, while monolingual corpora for Estonian has an order of 10^8 sentences and English has billions of sentences. There is about an order of 100 less parallel sentences.

- Modular pipeline for unsupervised machine translation. This means that pipeline is made up of different parts, which can all be improved, for example adding semi-supervision like adding a seed dictionary for translating words.
- Possibility to change languages more easily, compared to other systems, our system uses fewer resources, which lead to quicker deployment.
- Works for any language with sufficiently large monolingual corpora. The corpora should be sufficient to learn word and n-gram embeddings.
- Open source software and solution (git, e), which means that everything is available for the reader to explore and possibly to make further improvements or use it to make a translation system for languages that the reader is interested in.

1.4 Roadmap

In this section, we present how this thesis dissertation is organized. The thesis contains five chapters and they are as follows:

- The first chapter provides a general overview of the thesis.
- Chapter 2 contains related work. There have been 3 papers on Unsupervised MT and all of them are discussed.
- Chapter 3 is for technical background or methodology, where all of the pieces that are crucial for our solution are described. So, a broader theoretical overview of the practical components.
- Chapter 4 is for approach, where the pipeline design is described with additional details and available options.
- Chapter 5 is for experiments, firstly the setup and resources required are mentioned, then data and preprocessing is described. Choices for the solution, which is a pipeline, are given and a short description of why some library was used is also given. Finally, the setup of experiments is reported, with information about languages and test sets.

- Chapter 6 is for analysis of the experiments. Both quantitative and qualitative analysis is done for the results.
- In the end, there is a conclusion. Additionally, the possible improvements are mentioned.

2 Related Work

The solution for unsupervised MT has been the object of research for decades and finally in the late 2017 first promising works saw the light. To date, there are three unsupervised MT papers published and all of them will be analysed in this section. Before getting to those papers, however, one essential concept should be investigated, and that is back-translation, which is vital for the articles mentioned before. Back-translation is also relevant for semi-supervised and supervised NMT systems.

2.1 Improving Neural Machine Translation Models with Monolingual Data

The innovativeness from this paper (Sennrich et al., 2015b) comes from using monolingual data to enhance the quality of the translation. Until this paper, the state-of-the-art machine translation systems were neural and required a lot of parallel data. The authors look at different ways to make use of monolingual data. Their contributions are:

- Adding monolingual target sentences into the training set, which improves quality of the translation.
- Explore two ways to add monolingual training data to source side - firstly, using dummy source sentence, and secondly, using back-translation for source sentences.
- They use monolingual or in-domain parallel sentences to adapt NMT models for a new domain.

Using dummy sentences means that source sentence is empty, but target sentence is a valid sentence. This dummy method trains only the language model and sentence generation abilities of the target side, so while training, the weights of the source side encoder are frozen. They define back-translation as feeding monolingual training sentences with their corresponding synthetic source sentences, which are obtained by automatically translating the target into the source language. Here they mix synthetic parallel text with the parallel corpus to collect more data.
They evaluate on English-German dataset from WMT'15 and also report that parallel corpus has 4.2M sentences, but for English, there are 118M sentences, and for German, there are 160M sentences. They state that adding dummy sentences improves quality by 0.4 - 1 BLEU point and if synthetic data is added as well, then BLEU improves by 2.8-3.4 BLEU points over the baseline.

2.2 Unsupervised Neural Machine Translation

The motivation for the paper (Artetxe et al., 2017b) comes from the same place as the motivation for this work - shortage of sufficiently large parallel corpora for most language pairs, both low-resource languages (like Basque, where the authors are from) or pairs of main languages (like German-Russian pair). They developed three ways to do NMT - unsupervised, semi-supervised and supervised.
Their work is based on unsupervised cross-lingual word embeddings. Because the cross-lingual embeddings exist in the same space, the translation system can share the encoder for both source and target language and thus can be trained with only monolingual data. Noising is introduced to generate sensible target sentences and not just copy input to output. Noising is done by random token swaps. Back-translation is incorporated to further improve translation quality. Their system shares the same main components as any modern NMT system - encoder-decoder architecture with attention mechanism. Their system does have three novel aspects:

- **Dual structure** Traditionally NMT systems are built in one direction, e.g. from French to English. Their system handles both directions at the same time so English to French to English.

- **Shared Encoder** Dual structure is achieved by sharing encoder between the source and target language. The aim is to use the universal encoder to learn the representation of the input text independent from language and then each decoder should transform the representation to the corresponding language. That is a fantastic idea because the representation for sentence in any language is the same and then just a translation has to be generated in any language.

- **Frozen weights for embeddings in the encoder** Most NMT systems randomly assign embedding weights and update them during training, in this paper pre-trained cross-lingual embeddings weights were frozen, so they do not change iteratively in the training process. The encoder has to learn to build up the representation of the sentence from fixed embeddings. The embeddings are in the same space, but still separated, and thus, the word pool (meaning half in Estonian) would get a different embedding vector in Estonian and English for encoding. Similarly, word university and ülikool would get similar embeddings, because they are close in the vector space.

Using the three architectural changes, authors can train the entire unsupervised system following the following strategies:

- **Denoising** Thanks to the shared encoder and the dual structure, the system can directly be trained to reproduce its input. The system can be trained

by taking input sentence in the given language, encoding it using the shared encoder, and then producing the original sentence using the decoder. There are two decoders, one for each language. Resulting model should be able to produce a translation by replacing the decoders for target language decoder. This system is compromised, however, because the system learns to copy the input and not learn to capture the real knowledge. To avoid copying, random noise is introduced to the source sentence before translation, and the translation is compared to the real source sentence while training. Noise means here that random swaps are made to contiguous words, for a sentence with N words, N/2 swaps are made. Thus, adding noise achieves several goals, the system learns about the internal structure of the language and to restore the correct word order.

- **On-the-fly Back-translation** The denoising strategy still is a copying task with some synthetic changes but does not give good results for the final task of translating. Back-translation is introduced to train the system better - because the system operates dually, monolingual data can be translated from source to target language, generating a pseudo-parallel sentence pair. Training steps for the solution will predict the original sentence from synthetic translation. Different from standard back-translation, where entire corpus is back-translated at one time, authors use the dual structure allows doing back-translation while training.

Authors use the WMT'14 dataset - French-English and German-English. The authors train their system in three different settings:

1. **Unsupervised** The central scenario, where the system has access to only monolingual corpora.

2. **Semi-supervised** This is the case for many language pairs - instead of no parallel sentences, there is a small parallel corpus. In their semi-supervised systems, they used 10k and 100k parallel sentences, which are both not enough for state-of-the-art results with NMT.

3. **Supervised** The traditional scenario for NMT using lots of parallel data. Supervised NMT is not the focus but does give a reliable upper bound on how well the system can learn.

They compare the three different systems and additionally evaluate against a system that does nearest neighbour search for every word it sees, which they call a baseline. Their results show that their method works, and on French to English dataset, the baseline has 9.28 BLEU score, but their suggested unsupervised NMT has score 15.56 BLEU. Developing that baseline even further with 10k parallel data

improves BLEU in every test, and for French to English, the score becomes 18.57 BLEU. Adding more parallel data (100k dataset) improved results even further. One noteworthy thing is that the supervised system trained of full data did not perform as well as semi-supervised with 100k and the authors propose that is due to constraints put on the system. State-of-the-art systems on the same data have much better results.

2.3 Unsupervised Machine Translation Using Monolingual Corpora

The unsupervised NMT system that described in the article (Lample et al., 2017) is quite similar to the method described in the article above. The authors make one assumption for their machine - monolingual corpus exists for the source and target language. Similarly to the previous article, the fundamental concepts are following:

- Common latent space between the source and the target languages obtained by cross-lingual word embeddings.
- Model learns to reconstruct a sentence from a noisy source for given language.
- Model also learns to reproduce any source sentence from a noisy translation of that sentence, in both languages. Translated sentences are attained by doing back-translation.
- Initial model is a naive baseline model, which does a word by word translation using the cross-lingual word embeddings. Doing this encodes source sentences into the same latent space, from where decoding into any of the two target languages is done to get a translation.

Their system differs from the previous article by sharing a decoder as well, so their system has one shared encoder and one decoder, the difference comes from using a different lookup table for languages.
At a high level, their model starts with the unsupervised naive baseline. Then at every iteration, the system consisting of encoder and decoder are trained by minimizing an evaluation function, which measures the systems ability to reconstruct and translate from a noisy version. The noisy input is achieved by dropping and swapping words in the auto-encoding task while being itself a result of a translation from the previous iteration. The new system, which is trained for this iteration, is used for generating data for the system at the next iteration.
Additionally, at the iterative learning time, a discriminator is learned, which is a binary classification model, working on top of the encoder. In other words, the

discriminator says whether or not the encoding output is for the source or the target sentence, and is used for the final objective function, which evaluates the system.
Denoising is done differently in this paper. Firstly, a word is dropped from the sentence with some probability p, and secondly, they shuffle the input. Both of these strategies improve the translation system. Authors also report that without denoising, the system would learn only to copy source to target one-to-one.
The authors report various results, but for comparison with the previous model, let us consider only the WMT'14 English-French dataset. Their most important result is that unsupervised NMT achieves the same result as a state-of-the-art NMT system with around 100k sentences. They also suggest that BPE (Sennrich et al., 2015a) could further improve their results. The authors report the performance of their unsupervised NMT system at third iteration, which is 15.05 and 14.31 BLEU points (En-Fr and Fr-En respectively), which is about the same as the article previously. However, the system described in this section might have a higher upper bound for performance, since the same system trained with parallel corpora managed to get 28 BLEU points for supervised En-Fr translation.

2.4 Phrase-Based & Neural Unsupervised Machine Translation

There are many novel ideas in this paper (Lample et al., 2018) - they propose two different systems variants, phrase-based and neural model. Additionally, they continue forward from the ideas of the two articles above to achieve simpler models with fewer hyper-parameters, while achieving better results. The paper very nicely sums up the key aspects of the two previous works as:

1. Bilingual word embeddings are carefully initialized and derived using unsupervised methods, so the embeddings for source and target language exist in the same latent space.

2. Strong language models, which are trained using a denoising strategy, so the system learns to reconstruct sentence from a noisy source.

3. Back-translation enables turning the unsupervised task into supervised by generating synthetic translations to get parallel corpora and thus turning the task into a supervised task. For back-translation, the papers above and this paper also use dual models - translating from source to target and vice versa.

4. Same latent space for encoder output means that the encoder tries to express the meaning of the sentence independent from the language of the sentence

and then it is the job of the decoder to express that sentence in the target language, whichever may it be.

These four points are the cornerstones for unsupervised machine translation for all three articles. This paper combines the work of the previous neural approaches, simplifying the architecture and the loss function, which results in an easier to train and tune solution.
The same ideas and methods are applied to train a traditional phrase-based statistical machine translation (PBSMT) system. PBSMT systems are useful if data is scarce, which is the case for unsupervised MT because they count occurrences of the phrases, but neural solutions typically try to leverage vast amounts of data to fit hundreds of millions of parameters to attain distributed representations. Neural methods may generalize better with abundant data, but risk overfitting with limited data. The obtained PBSMT model is simple, effective, straightforward, novel, easy to interpret and fast to train.
Authors claim that unsupervised MT can be achieved by leveraging the three following key components, which are used in both the NMT and PBSMT systems,

- **Initialization** When a bilingual dictionary is inferred in an unsupervised fashion, baseline model can be a simple "word-by-word" translation system.
- **Language modeling** Given lots of monolingual data, a language model for source and target languages can be trained. Language models express how good a sentence is, and can be used to evaluate the goodness of a translation. In this paper, the language model is obtained by denoising autoencoding.
- **Iterative back-translation** The most effective way to make use of monolingual data. Back-translation generates training data for the models that are trained on the next iteration. The first iteration uses the baseline. Thus, the unsupervised task becomes a supervised task, by providing a "noisy" source for a real target sentence.

One novel thing about this paper is that bilingual dictionaries are no longer used for related languages, unrelated languages still use bilingual dictionaries. The authors make use of byte pair encoding (BPE) for related languages, they replace words and have the following advantages: reducing vocabulary size by eliminating the presence of unknown words for target language and instead of a mapping, BPE tokens are defined by jointly processing both corpora together. In practice, this means that:

1. Source and target corpora are joined together
2. BPE tokenization is applied on the joint corpora

3. Token embeddings are learned on the joint corpora

Their system also shares the encoder and decoder and mention that sharing the decoder is critical, but sharing a decoder introduces regularization. For the decoder, the first token specifies the language of the target output. This architecture is quite similar to the previous two works, but compared to (Artetxe et al., 2017b), the decoder is shared. Compared to (Lample et al., 2017) online back-translation is used, and the adversarial term is removed, thus simplifying the loss function. These changes make the system simpler and reduce the number of hyper-parameters.
The unsupervised PBSMT has almost identical structure, but still minor changes should be reported. Traditional PBSMT systems use bilingual data to populate the phrase tables, with monolingual data. However, it is not apparent how to generate the phrase tables. The language model can be trained on monolingual data. The phrase tables are populated by using dictionary extraction methods described before (bilingual embeddings), which give us a baseline model. The phrase table initially has only unigrams or words inside it. Iterative back-translation helps to improve the phrase tables further, by generating bigger tables, consequently making PBSMT iteratively better.
The dataset they evaluate their results contain four language pairs, common English-French and English-German, but also less common English-Romanian and English-Russian. For comparison, here, only English-French (WMT'14 dataset) results are reported. The results show that both of their new methods significantly outperform the previous results. The performance increase is about 10 to 12 BLEU points, which is enormous. The PBSMT method gets 27.1 BLEU points for En-Fr and 24.7 for Fr-En. The NMT gets 25.1 and 24.2 BLEU (En-Fr and Fr-En respectively). The authors additionally try a combination of PBSMT and NMT and claim that in general case, these perform best (and the data backs it up).

3 Technical Background

In this section, we define or explore the important background that is necessary for the solution. The technical knowledge is organized in the same way as the modular pipeline, so starting from data processing and ending with how to get translations.

3.1 Phrases or n-grams

The motivation for using phrases comes from the fact that words are ambiguous, and phrases provide more context and are less ambiguous, drawback, however, is the explosion of the number of words. For example in Estonian, we have words "laual" and "laua peal", which mean the same thing("on the table"), but one is a bigram, in English an example would be "it is" and "it's".
For extracting n-grams, a probabilistic joining algorithm is run (Tättar and Fishel, 2017). The probabilistic joining algorithm process text, so that both words and n-grams are selected for the output. The reason for calling these values n-grams and not phrases is that we cannot be certain, that the output has only meaningful phrases.
The goal is reached by doing frequency filtering with sampling, which sometimes includes or excludes the n-gram based on probability. The n-gram probability is a smoothed reverse frequency, which downsamples more frequent words, it is calculated with formula $p = \frac{1}{f^\beta}$, where p is the sampling probability, f is the n-gram frequency and β is a small weight. For example, a bigram with frequency 20, and $\beta = \frac{1}{8}$, the probability of sampling is 0.688, but for bigram with frequency 500, the probability of sampling is 0.460.

3.2 N-gram embeddings

Embeddings for n-grams are trained exactly like the embeddings for words because they are single entities. The motive is that the cosine similarity for the vectors of n-grams "it's" and "it is" would be the same or almost the same.
There are two primary ways to learn word embeddings - skip-gram model and the continuous bag-of-words (CBOW) model - both of them are log-linear methods (Mikolov et al., 2013a).

- Continuous bag-of-words **(CBOW)** method tries to predict the word from the context of surrounding words. It is called a bag-of-words model because the surrounding words have no structure and are put into a data structure called bag-of-words (BOW). To sum up, the training goal is to correctly predict the word from its context, which is a collection of words called BOW.

It is called continuous because continuous distributed representation are used for the context instead of just words.

- The second architecture called **skip-gram** is very similar to CBOW, but instead of predicting the current word based on context, the model tries to predict the surrounding words from the current word (the input). Surrounding words mean that n words before and after, where n is a parameter. Bigger n makes the word embeddings better, but the training complexity also increases.

The models described above for word embeddings have one disadvantage, and that is the word level nature if the model has not seen some word in the training data, there are no embeddings for that word. This is a problem for morphologically rich languages like Estonian, which has 14 cases for nouns. Although the words are different, they differ only by few characters and that idea is the basis for enriching word vectors with subword information (Bojanowski et al., 2016). Subword here means that the characters which make up a word are used to get word embeddings in the form of character n-grams of varying size. Their work is an enhancement of the word-level model. The subword model represents a word as a sum of its character level n-grams. One clear advantage of the character level enriching is that it enables to get vectors for out-of-vocabulary words, which is especially helpful for language like Estonian. Authors additionally show that their method is the new best way to find embeddings.

3.3 Cross-lingual embeddings

There has been a lot of work concerning the bilingual word embeddings, which are described in the following survey (Ruder, 2017). So far the most work has done in a supervised manner, but recent work has done for getting cross-lingual embeddings in a semi-supervised manner, and completely unsupervised methods also exist now. The work on semi-supervised and unsupervised methods sparked the research interest for unsupervised NMT. The supervised versions use seed dictionary, and a rotation matrix, that maps the two embedding spaces (Mikolov et al., 2013b) and (Xing et al., 2015) suggests a novel idea of using an orthogonality constraint for the rotation matrix W, which makes the task have a closed form solution. The drawback here is of course that what if there is no seed dictionary. There are also semi-supervised methods which take the seed as input like supervised versions, but the seed can be obtained in almost any case, like using numbers to find the mapping of the spaces (Artetxe et al., 2017a) or strings with identical characters as the seed (Conneau et al., 2017). Semi-supervised methods also imply some knowledge about

the language pair. The unsupervised methods use no seed dictionary and learn from just vectors trained on monolingual data. The unsupervised paper for "Word translation without parallel data" (Conneau et al., 2017) had terrific results, and the authors reported unsupervised word translation results better than previous best results, which were supervised. The method takes word embeddings in source and target language as inputs. The objective is to find the best mapping matrix W, which is calculated by singular-value decomposition. The hard part is finding the initial mapping, which acts as the anchor points for finding the mapping. The training is happening like a two-player game, with two players:

- Discriminator aims to maximize the ability to predict the language of the embeddings after mapping.
- Mapping matrix seeks to minimize the discriminator's success, by making itself better.

To make this mapping iteratively better, a synthetic dictionary is learned by the mapping matrix W, and only the most confident and frequent words are kept in the mapping. Then the system is trained again using an improved Procrustes method, but these iterative steps don't have a significant performance impact since the initial mapping is already quite strong. Authors also develop one metric for evaluating closest words, that relieves the effect of hub-words in high dimensional spaces, which provide much better translation accuracies.
Considering the common motif of assessing on English-French language pair, the authors report 82.3% precision for top 1-word mapping compared to the supervised method having precision 81.1%. Precision here means that after mapping only one word is compared to a test dictionary, so one-to-one mapping. The nearest neighbor has accuracy 78.1% compared to the improved metric. For French-English, the supervised method gets 82.4%, while unsupervised method gets 82.1%.

3.4 Neural language model

Language model in simplest terms tries to predict the relative likelihood of the input sentence. There are phrase based, statistical and neural language models. Since 2010, everyone has started using recurrent neural language models (Mikolov et al., 2010). The common way to train language models is to train one model for every language. The authors of the paper "Continuous multilinguality with language vectors" (Östling and Tiedemann, 2017) described a way to use different languages in a single model.
The motivation about having all languages in one system is that languages are related and share many features together, this fact is ignored with using one-language models. Secondly, for many languages there are no huge corpora like for

English, we can let the algorithm learn relations between languages. The way the authors separate different languages is to give a one-hot vector as the first element of a sequence, which describes the language to use.

3.5 Effective searching through exponential search space

Algorithm 1: Pseudo-Code for Beam Search algorithm. Source: (Koehn, 2010)

```
1  place empty hypothesis into stack 0
2  foreach stack 0...n-1 do
3      foreach hypothesis in stack do
4          foreach translation option do
5              if applicable then
6                  create new hypothesis
7                  place in stack
8                  prune stack if too big
9              end
10         end
11     end
12 end
```

The search space for possible translations is exponentially growing, which means that the task of finding the best translation is computationally costly. Consider this example: source sentence has 10 words (n), there are 10! ways to reorder the words and for 5 different ways to translate one word(t), which means the complexity is actually $O(t * n!)$, which is actually factorially growing search space. The optimal solution using a dynamic programming algorithm does make it an exponential searching space, but it is no help because translations need to be fast. This problem is alleviated by using a clever, but greedy, search algorithm called the beam search, showed on Figure 1. Beam search is an extension of the breadth-first search, where at each level of the tree, all children are generated, but only k children nodes are explored, the others are pruned out because they are likely to lead us to unlikely/bad states. Increasing k means that the beam is wider and it is more likely that the best solution is found, however, this does require more computing power, on the other hand, decreasing k means that the best solution might be not that good. It is important to note that with infinite beam width the algorithm reverts to being the breadth-first search. Usually, the first end state is returned for beam search.
The algorithm is shown on Figure 1. First, the empty hypothesis created, which

is placed in the first stack. Stacks show the number of source words translated, so hypothesis with zero words translated goes into the first stack, the hypothesis with three words translated go to the third stack. We iterate over the stacks, starting from first, and for each hypothesis, we additionally generate all translation possibilities and make the new hypothesis, if applicable. The new hypothesis is placed into the stack, where we first try to combine with the previous hypothesis and then if the stack is too big (bigger than beam width k), we prune the stack by throwing away the least likely option.
The number of stacks is equal to the number of words in the source sentence, let's call it sentence length n. The number of hypothesis in a stack is limited by the beam width k. The number of options for the new translation is limited by the number of source words. So the complexity is the following:
$O(\text{sentence length} * \text{beam size} * \text{number of translation options})$, but since the number of translation for a hypothesis has a linear relationship with the sentence length, the complexity is: $O(k * n^2)$.

4 Approach

The approach I worked out for unsupervised NMT is based on unsupervised cross-lingual word embeddings for dictionary extraction. The method is built for modularity, so every module could be switched out easily because when something could be done better, it would be beneficial to switch that part out. The pipeline is made up of 4 parts:

- **Phrase (N-gram extraction)** The point of using phrases is that they should be less ambiguous than words, and the fact that one word in one language can be a phrase in another language, like Estonian word "laualt", which means "from the table". The extraction of n-grams is done by an algorithm we developed (Tättar and Fishel, 2017) and described in the previous section. The code is open source and freely available (git, c). The Bleu2vec method has been modified to use FastText (git, d) word embeddings instead of word2vec, so only the phrase extraction part is used. FastText embeddings are better than word2vec because they incorporate subword level info.

- **Unsupervised Cross-lingual phrase embeddings** After finding vectors for phrases, we need to project the source and target language embeddings into the same space. Projecting is done with the MUSE library (git, b). MUSE works with FastText embeddings and is developed by the authors of the article "Word translation without parallel data" (Conneau et al., 2017), which was described in Section 3.3. MUSE learns the mapping of source phrases to target phrases, and after training, we can do the nearest neighbor search for the source word from the same space as the target n-grams. The nearest neighbors are found by cosine similarity score, that goes from 0 to 1, where 1 is very similar and 0 means not similar at all.

- **Language model** The language model of choice uses RNN cells, more specifically LSTMs. The reason for going neural is that neural is more fluent than statistical or phrase based, but does require more computational power, usually running with GPU-s. The motivation behind a neural language model is that we saw the output of the first translations were very robust and a measure of goodness is required for the hypothesis that is generated. The language model I used is available on GitHub(git, a), called CatLM, which is a categorical character or word based neural language model, which can learn different domains or categories(like different languages) for the language to generate.

- **Beam search** I implemented my own beam search based on the pseudo-code on the Figure 1. Additionally, the beam search algorithm looks for all

the n-grams in the dictionary and then for every input n-gram of length up to n are generated, so unigrams, bigrams, and trigrams. After getting the input n-grams, the translations are calculated, and k nearest neighbors are selected as the translations. Then the normal pseudo-code is followed, and before the new hypothesis is generated, the unfinished translation is scored with the neural language model. After adding the new hypothesis to the stack, the stack might need to be pruned, if there are more elements than the beam width allows. The stack is ordered by negative log-likelihood. The negative log-likelihood is made up of two parts:

1. language model score for the hypothesis, which is the negative log-likelihood score,
2. the sum of taking the logarithm of translation options similarity score. These logarithms of similarity scores are negative because the scores are between 0 and 1.

In the end, the final beam with k (beam width) translations is returned.

4.1 Motivation for my approach

In the related work section, I described the current state-of-the-art of unsupervised machine translation. They all had baseline systems and then the models were iteratively improved using back-translation. The reader of this thesis might be asking, why there is no back-translation in my method. The answer is quite simple - lack of resources, one batch of back-translation reportedly takes a couple of days of GPU time, which I simply don't have.
All of the related work used the naive baseline - word-to-word translation. I aim to improve the baseline model to get better translations on the first iteration, which might be able to make the iterative methods converge more quickly. Of course, it would be nice to get the same performance as the phrase-based statistical machine translation (Lample et al., 2018), but that was also iteratively improved.

4.2 Differences compared to Statistical machine translation

The reader might wonder that this method seems awfully similar to statistical machine translation (SMT). Indeed, this method is similar to SMT, just that the extensive phrase tables have been switched out with translation table acquired by using n-gram embeddings. SMT algorithms like MOSES (mos) have no way of getting phrase tables in an unsupervised fashion, they cannot do the alignment correctly. Another difference is that the n-gram language model incorporated to

SMT is switched out by recurrent neural language model, which promises better fluency for translations. The final translation algorithm - beam search, is a modified version of the search algorithm used in SMT.

5 Experiments

I first describe the datasets, then the setup and parameters and finally I compare my method to the word-to-word naive baseline. The code to reproduce these results is available on GitHub (git, e).

5.1 Data

In my experiments, the datasets are from the WMT'18 unsupervised translation task (wmt). Only monolingual datasets are used. Only Estonian - English and English - Estonian language pairs are considered for validation sets. The domain of the data is news.
There are two types of preprocessing for my method:

- **Bilingual embeddings** First n-grams have to be extracted, and all punctuation is removed, so alphanumeric characters remain. The data is then lowered.
- **Language model** is trained on 15 million sentences, tokenized by Moses and then lowered. The punctuation is kept, for making the output smoother, by learning the punctuation.

The validation dataset is made up of 2000 sentences for both language pairs.

5.2 Setup

In this section, I describe the parameters I used for my models:

- **Naive Baseline** The baseline model is the same as in the related work - word-to-word translation. The baseline has unigram cross-lingual word embeddings, it does not look at the n-grams. This is done in order to be consistent with related work.
- **N-gram extraction** Frequency filter settings are the following: 20, 120 and 90 (frequency counts for filtering out infrequent n-grams, unigrams, bigrams and trigrams respectively). The beta parameter is set to 0.125.

- **FastText Embeddings** Mostly default parameters, the method used is CBOW with character n-grams of size 3 to 6. The number of dimensions is set to 300. Other parameters are kept as default.

- **Bilingual MUSE embeddings** Default parameters are kept and MUSE is used on CUDA GPUs. Embeddings dimensions are set to 300. I tried to change parameters and the number of refinement steps, but they didn't yield any positive results.

- **Language model with CatLM** - I try two different language models - word based and character based models. There are pros and cons for both approaches:

    1. **Word level LM** The sequence of up to 30 words and vocabulary of size 40000. The problem with word-based language model is that there might be many words, which don't fit into the 40000 vocabulary size, but this should not affect performance much. The advantage is that a sequence of 30 entities can be evaluated. Too big vocabulary size with long sequence makes the training weights too big. Otherwise, they could be even larger.
    2. **Character level LM** The obvious advantage of a character level language model is that there are no unknown words, if preprocessing is done, then the vocabulary size should be less than 100 with no unknown characters. The disadvantage is that the sequences must be long, which make the weights not fit into the memory of a GPU. The longer sequence makes the evaluation of a hypothesis more complex and takes more time, and the character sequences must be very long. For the sake of efficiency I used 150 characters, but even if every word with space had 5 characters, this would still mean a sequence of 30 words. Unfortunately, the sentences to be translated are often longer than 150 characters, and they get partial translations.

- **Beam search** I use two basic parameters for beam search - stack size k and number of translation options n. Increasing any of the parameters makes the search more complex and time-consuming. The parameters I use for all translations is $k = 5$ and $n = 3$, which means that every source word gets three best translation options and five current best hypothesis are kept for every stack.

5.3 Example

Let us consider the following sentence "See on väike test" and let's translate it. The translation options I get with MUSE are shown on Table 1.

Table 1: Translation table

Source n-gram	Target top 3 n-ngrams			Similarity scores		
see	it	it__it	this__it	0.826	0.821	0.802
on	is	is__is	is__are	0.927	0.890	0.827
väike	small	small__little	small_c	0.709	0.670	0.670
test	test__that	one__test	test__them	0.702	0.686	0.684
see__on	it__is	this__is	is__it	0.875	0.861	0.854
on__väike	is__small	small__is	is__modest	0.756	0.702	0.666

The translation table is clearly not perfect and favors bigrams for english language, so unigram in Estonian and the translation is often a bigram in English. The following 5 sentences are the top 5 translations for the sentence "see on väike test":

1. "small is__it one__test"
2. "it small__is one__test"
3. "small it__is one__test"
4. "it is__small one__test"
5. "it__is small one__test"

Figure 1: Top 5 translation options generated by beam search

When looking at the sentences generated, we can see that they are not perfect. The faults from dictionary propagate into the translation. The perfect translation would be "this is a small test", but the best translation I can come up that consists of the n-grams given is "it is__small one__test". So this is a hard task, to put together a good sentence from imperfect translations. For some reason, the repeating unigrams are often translation options like "that__that" instead of "that" being the top choice.
The baseline model translates the same sentence into "it is small diagnostically .". The baseline here is lucky, because it does word-by-word translation. If the word order had changed, baseline model would have done even worse. The translation

is almost ok, just the word "test" is translated as "diagnostically". The example given is generated with the parameters given in this section.

6 Results and analysis

This section is for the results. The results are not entirely optimistic. The BLEU score for Estonian to English for 1867 sentences is 0.8 for the naive baseline and 0.98 for my method. This might imply that my method is slightly better, but these results don't give info about how it translates, because BLEU less than 1 is not good. I didn't have enough translations for English to Estonian, to report any meaningful BLEU score.

6.1 Qualitative analysis

6.1.1 Estonian to English

To better understand the results, I carry out qualitative analysis. I pick 50 random sentences and do human evaluation for the sentences. Categories are:

1. **Good** The meaning of the sentence is carried on.
2. **Ok** The meaning is ok, but there are mistakes.
3. **Bad** Some keywords are there, but overall a bad translation.
4. **Horrible** If you try enough to think about the sentence, then you understand that the translation and reference are related.
5. **Random** Translation seems like a random set on words.

Results are shown on Table 2. It does mostly horrible and random translations, but sneaks in some good translations as well. Compared to the baseline model, the output looks better for the system I developed. Let us look at some examples.

Table 2: Results for translating English to Estonian

good	1
ok	1
bad	2
horrible	10
random	6

Good examples come basically from good dictionary - 1 to 3 word sentences, that are translated correctly, like

- Source: Väldi kohvi.
- Reference: Avoid coffee.
- Translation: Coffee avoid.

The ok looking translations are shorter sentences, like

- Source: See on lahendamata küsimus.
- Reference: That's an unsettled question.
- Translation: This is fundamental question.

The bad translations translate some words wrongly, mostly because the dictionary entries are like that, but the output might be a valid sentence.

- Source: Ma olin vihane ja nutsin.
- Reference: I was angry and crying.
- Translation: So was my bed angry.

A second longer example for bad translation:

- Source: Nagu juba öeldud: vesi ja mahl on su sõbrad.
- Reference: As already said, water and juice are your friends.
- Translation: Like as that stated: water is and your friends juice.

There are a lot of terrible and out of context sentences. Some parts might be even good, but for long sentences the errors propagate. These sentences are very hard even for state-of-the-art systems trained on huge parallel corpora. I provide some longer examples now for horrible sentences:

- Source: Suurbritannia politsei on öelnud, et nad usuvad, et Londoni Grenfell Toweri isolatsioon ja fassaadipaneelid võisid panustada seal juunis toiminud tulekahju kiiresse levikusse, milles hukkus ligikaudu 80 inimest.
- Reference: In the uk, police have said they believe the system of insulation and cladding panels on london's grenfell tower may have contributed to the rapid spread of a fire there in june in which some 80 people died .
- Translation: Police is is told that, that they believe, Barcelona and that the fast invest the been to the sonic insulation the spread here then fire damage even existed in october, seven people wherein the killed was killed approximately four.

We can see obvious mistakes here, like London translated into Barcelona, numbers translated into different numbers, fluency/adequacy mistakes and repeating word mistakes. The repeating word mistakes are very problematic especially for me (the other mistakes happen in any machine translation system), because they come from imperfect dictionary, so the word "öelnud", which is translated into "is_ _said", while "on" is translated to "is", thus the translation becoming "is is said". There are many examples like these, but I believe that this sentence kind of shows all the problems I have.
There are also random looking or even hysterical looking sentences. I provide one here:

- Source: Peoliste melu naist ei sega.
- Reference: The festivalgoers' revelry doesn' t faze the woman.
- Translation: Don man don men she bother with.

This translation sounds horrible and makes no sense. The problem is with the tokenization actually, the word "ei" is translated as "don", which comes from "don't", tokenized as "don" and "'t". The second problem is again the imperfect dictionary, as "peoliste melu" is translated as "men don".

6.1.2 English to Estonian

The problems described previously in Estonian to English translation are present here as well and the distribution for the quality of the translation remains the same. For Estonian I tried both word and character based models, about 80 sentences (technical constraints), which are not enough to report confident BLEU, but qualitative analysis can be done. A couple of examples:

- Source: It was a special moment for me.
- Reference: See oli minu jaoks eriline hetk.
- Translation: See ühe suurhetk mul tõesti.

And one more:

- Source: Although challengers tried to crowd out the four-time olympic champion in the final lap, 34-year-old farah dug deep on the home stretch and won with a time of 26:49.51.

- Reference: Kuigi viimasel ringil proovisid konkurendid neljakordset olümpiavõitjat rajalt minema trügida, leidis 34-aastane Farah lõpusirgel jõudu ning võitis ajaga 26.49,51.

- Translation: Poolfinaal siiski ümber ja koguni juubeldada üritanud tiitlivõitja kaheksandikfinaal ning väljaspool ringiaeg, ja ühe kodusaali võitnud jala ning aega aega ning sügava ja sügavike kauge sugulane ikute.

6.2 Why the system does not work?

There are many reasons for the system to not work properly:

- **Inconsistent preprocessing** I initially thought that having different preprocessing for word/n-gram embeddings and language model would be fine, but in practice it seemed bad. Commonly, word embeddings are trained on data without punctuation and I did that, but at translation stage it was hard to remove all punctuation, without affecting meaning. This led to different problems while translating, especially for English as the source language. Problems were with words like "don't", "it's" and so on.

- **Slow language model** The problem with the current solution is that the language model is not very efficient. The idea behind the language model is good, it could be optimized for better performance. Language model hindered my translating process to couple hundred sentences per day. 99.98 % of the time for the algorithm was spent with the language model. I failed to optimize this step. Without a language model, the solution might as well fall back to naive baseline, as there is no way to estimate the goodness of a reordered target sentence.

- **Unrelated languages** One problem that might cause the system to produce bad translations is that the languages are completely unrelated. The dictionary qualities are different, for English-French/German language pair, the reported accuracies are between 70% to 80%. My dictionary has top one accuracy of around 24-27%. The accuracies are given on Table 3.
  One additional problem is the gender problem. In Estonian, there is no gender for pronouns, but the English language has gender based pronouns like "he" and "she".

Table 3: Word translation precisions, P means precision and number means how many candidates are looked at. P@5 means that instead of one candidate, 5 best candidates are considered. The paper referenced in the table is from paper (Conneau et al., 2017)

Pair	Reported by	P@1	P@5	P@10
En-Fr	paper	78.1	-	-
Fr-En	paper	78.2	-	-
En-De	paper	74.0	-	-
De-En	paper	72.2	-	-
En-Et	me	27.1	42.8	48.4
Et-En	me	24.0	38.3	43.9

7 Future work

The following updates could improve the system:

- Switching to the phrase-based statistical machine translation, where I use my generated method for extracting initial phrasetables. This method does seem promising.
- Making the beam search more efficient, by allowing only word changes that happen in its vicinity (like 4-6 words away).
- Out-of-vocabulary words have to be dealt with, one way would be to use BPE or assign the word itself as its translation.
- The problem currently is a slow language model that hinders the translation part of the system, so some alternative should be investigated.
- Switching the nearest neighbor from cross-lingual embeddings for translating words for the improved version called "CSLS" (Conneau et al., 2017).
- Switching the language model for some faster model or optimizing it.
- The method might work better for related languages and that should be tested.

8 Conclusion

The goal of this thesis was to achieve better results than the baseline, but this goal was not reached or we cannot say safely that it was reached. There are a lot of improvements one can make to the system, which could make it better, but I propose that the system architecture should be looked over and improved.
This work is a good base to continue working on unsupervised machine translation as the related work is thoroughly worked through. I have my base system and a lot of options for future work. Based on that I can consider this thesis a success even though the results are not good.

References

Catlm implementation (categorical language model) on github. https://github.com/TartuNLP/catlm, a. Accessed: 2018-05-21.

Muse implementation on github. https://github.com/facebookresearch/MUSE, b. Accessed: 2018-05-21.

Bleu2vec open implementation. https://github.com/TartuNLP/bleu2vec, c. Accessed: 2018-05-21.

Fasttext implementation on github. https://github.com/facebookresearch/fastText, d. Accessed: 2018-05-21.

Author's GitHub for unsupervised MT. https://github.com/crypotex/unsupervised_MT, e. Accessed: 2016-05-12.

Webpage for moses library. http://www.statmt.org/moses/. Accessed: 2018-05-21.

Webpage for the wmt'18 conference. http://www.statmt.org/wmt18/. Accessed: 2018-05-21.

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 2017a. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-1042.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. *CoRR*, abs/1710.11041, 2017b. URL http://arxiv.org/abs/1710.11041.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016. URL http://arxiv.org/abs/1607.04606.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *CoRR*, abs/1710.04087, 2017. URL http://arxiv.org/abs/1710.04087.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010. ISBN 0521874157, 9780521874151.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043, 2017. URL http://arxiv.org/abs/1711.00043.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. 2018. URL https://arxiv.org/abs/1804.07755.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, *INTERSPEECH*, pages 1045–1048. ISCA, 2010. URL http://dblp.uni-trier.de/db/conf/interspeech/interspeech2010.html#MikolovKBCK10.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013a. URL http://arxiv.org/abs/1301.3781.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013b. URL http://arxiv.org/abs/1309.4168.

Robert Östling and Jörg Tiedemann. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 644–649, Valencia, Spain, April 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E17-2102.

Sebastian Ruder. A survey of cross-lingual embedding models. *CoRR*, abs/1706.04902, 2017. URL http://arxiv.org/abs/1706.04902.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015a. URL http://arxiv.org/abs/1508.07909.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709, 2015b. URL http://arxiv.org/abs/1511.06709.

Andre Tättar and Mark Fishel. bleu2vec: the painfully familiar metric on continuous vector space steroids. In *Proceedings of the Second Conference on Machine Translation*, pages 619–622, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W17-4771.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011. Association for Computational Linguistics, 2015. doi: 10.3115/v1/N15-1104. URL http://www.aclweb.org/anthology/N15-1104.

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Andre Tättar**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   of my thesis

   Unsupervised Machine Translation using Cross-Lingual N-gram Embeddings

   supervised by Dr. Mark Fishel

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 21.05.2018