Targeted Sentiment Analysis for Norwegian text

Insights from NoReC$_{fine}$ for resource-rich, low-resource, and zero-shot scenarios

Egil Rønningstad

Thesis submitted for the degree of
Master in Language Technology
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2020

Targeted Sentiment Analysis for Norwegian text

Insights from NoReC$_{fine}$ for resource-rich, low-resource, and zero-shot scenarios

Egil Rønningstad

Targeted Sentiment Analysis for Norwegian text

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

Targeted Sentiment Analysis attempts to extract sentiment targets and the sentiment polarity towards these targets, as explicitly expressed in text. Targeted Sentiment Analysis is a difficult task where there may be multiple sentiment targets in one sentence, and there may be conflicting sentiments towards one target. Sentiment may be expressed through nuances and combinations of words at different positions in the sentence. State-of-the-art models for Targeted Sentiment Analysis therefore require large amount of data. In our thesis we explore approaches to Targeted Sentiment Analysis in scenarios where a) we have a large annotated dataset, b) we have a very limited amount of annotated data, and c) we have no annotated data for the target language and domain. Given a large monolingual dataset, we provide a state-of-the art model through the multilingual BERT (M-BERT) pretrained language model. Given more limited data we show how bilingual training data allows for noteworthy improvements over monolingual training. Given a scenario with no labeled data for the target domain and language, we demonstrate the cross-lingual performance of M-BERT for the Norwegian and English language pair. We isolate and compare the effect of domain and language differences, and demonstrate the option of machine-translating text for Targeted Sentiment Analysis.

Acknowledgements

I would like to thank my supervisors, Jeremy Barnes and Lilja Øvrelid in the Department of Informatics at the University of Oslo.

In appreciation of my father who taught me to always consider what would be the right tools to employ for the various work in our farm's repair shop.

A warm Thank you to my family who supported me during the writing and contributed with valuable proof-reading and feedback.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sentiment Analysis is a research field within Natural Language Processing (NLP) which seeks to determine people's opinions or sentiments, expressed in text. Are people positive or negative? What exactly are they positive or negative about? Sentiment Analysis is used commercially, to understand public opinion towards products and events, and can also be used politically to understand sentiment towards public figures or political parties and questions (Bakliwal et al. 2013; Hsieh et al. 2016; Wang et al. 2012). Sentiment Analysis is also used in the design of conversational agents, chat-bots, to assess how well the conversation is addressing the customer's need (Martins et al. 2020).

Sentiment Analysis may focus on classifying the overall sentiment for a document or sentence, i.e., whether it is positive or negative. The documents we analyze may be newspaper articles, reviews written by professionals, tweets, user-submitted reviews, etc. With reviews, the entire text is usually about a given entity, and document level Sentiment Analysis aims at detecting whether the text in total is positive or negative towards the entity being reviewed. But Sentiment Analysis can also focus on more fine-grained information about the sentiment inside each sentence.

Targeted Sentiment Analysis aims to detect each *sentiment target* explicitly mentioned in the text, along with the *sentiment polarity* towards the target. In the following sentence from a restaurant review, we find mixed sentiment, both positive and negative:

(1) *I liked the atmosphere very much but the food was not worth the price.*

The author of the sentence is positive towards *atmosphere*, and negative towards *food*. These are the *targets* for each expressed opinion, or sentiment. The task is to identify these opinion targets and the *polarity*, positive or negative towards each opinion target, or *sentiment target*. This analysis gives a more granular insight into the expressed sentiments. This can be helpful for those being reviewed, to understand what in particular needs to improve in order to get more positive reviews.

Resource scenarios for Targeted Sentiment Analysis Creating a dataset for Targeted Sentiment Analysis is a time-consuming annotation process

that requires skilled personnel. Sentiment expressions and targets need to be identified according to specific annotation rules. Datasets for Targeted Sentiment Analysis are therefore not so common and not so large. $\text{NoReC}_{\text{fine}}$ (Øvrelid et al. 2020) is such a dataset that contains Norwegian review texts manually annotated for sentiment targets and polarity. This allows for training models for Targeted Sentiment Analysis in Norwegian. The size and granularity of the annotations in this dataset makes it a comprehensive resource for fine-grained Sentiment Analysis, and not many languages have this resource available. With this dataset we explore three important resource scenarios:

a) In a resource-rich scenario we fine-tune models for Targeted Sentiment Analysis with the relatively large dataset $\text{NoReC}_{\text{fine}}$. We present a new state-of-the-art system for Targeted Sentiment Analysis in Norwegian, based on this dataset. We investigate the use of cross-lingual methods to further improve these results.

b) For a resource scenario with limited annotated data, we show how bilingual training data allows for noteworthy improvements over monolingual training. We present a system trained with 400 labeled sentences in the target language. This performs better than previous systems trained on more than 8000 sentences.

c) For a resource scenario without any labeled data in the target language and domain, we explore the cross-lingual and cross-domain ability of M-BERT, the main pretrained language model we use. We compare the benefits of cross-lingual data with cross-domain data.

New tools in the toolbox Systems for Targeted Sentiment Analysis have for some years been developed using deep neural networks. Architectures with Long Short-Term Memory (LSTM) as a core component were until recently dominating on leaderboards (Yang et al. 2018). This is also the architecture used for the initial baseline experiments published with the dataset (Øvrelid et al. 2020). During the writing of our thesis, Transformer-based models like BERT have been employed for various NLP tasks, and they often outperform LSTM-based models (Rietzler et al. 2020). We experiment with versions of BERT, both monolingual English versions and the multilingual BERT (M-BERT) that has Norwegian in its training data. We explore how a BERT-based system compares with LSTM-based systems, and how bilingual training data can improve results, compared with monolingual training data.

Our experiments are based on the following research questions:

RQ1: Which neural architecture is best for creating a model for Targeted Sentiment Analysis in Norwegian, based on the $\text{NoReC}_{\text{fine}}$ dataset?

1a) Are the popular LSTM-based models with pretrained word embeddings still the best choice for Targeted Sentiment Analysis on Norwegian texts?

1b) Are there Transformer-based models available for Norwegian text that may be able to outperform LSTM-based models here, like they often do with English NLP tasks?

RQ2: Are there any English resources that can improve our Norwegian model for Targeted Sentiment Analysis?

2a) Can English training data be added to the Norwegian, in order to improve our model?

2b) Can we use machine-translation to move our texts from Norwegian to English, and get better results there?

RQ3: $\text{NoReC}_{\text{fine}}$ is a comparatively large dataset. What can be done when this amount of training data is not available?

3a) Which of the tested methods might help if the available training sentences are in the hundreds, and not in the thousands?

3b) If there is no training data from the same domain and language, to what degree may data from other domains and languages be useful?

1.1 Overview

Chapter 2 provides further background on the NLP task of Targeted Sentiment Analysis, both historical developments and previous work. Pretrained language models are introduced, both monolingual and multilingual, as well as LSTM-based and Transformer-based systems for Targeted Sentiment Analysis.

Chapter 3 presents the dataset, $\text{NoReC}_{\text{fine}}$. We compare it with an English dataset and look at what makes $\text{NoReC}_{\text{fine}}$ unique, both as a resource and a challenge for Targeted Sentiment Analysis.

Chapter 4 presents our experiments with $\text{NoReC}_{\text{fine}}$, finding the best model for Targeted Sentiment Analysis based on this training data.

Chapter 5 presents bilingual experiments with $\text{NoReC}_{\text{fine}}$. We test adding English training data, and test a machine-translated version of the Norwegian dataset.

Chapter 6 presents our experiments with reduced versions of the dataset. $\text{NoReC}_{\text{fine}}$ has more than eight thousand sentences in the training set. This is relatively large for a labour-intensive dataset like this. We show which of our approaches from previous experiments can be helpful when there is less available training data.

Chapter 7 presents cross-domain and cross-lingual experiments where there are no labeled data from the same domain and language as the testing data.

Chapter 8 summarizes the thesis and presents suggestion for further work.

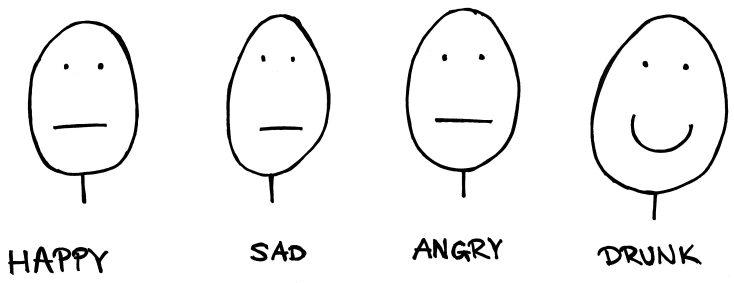


Figure 1.1: Norwegians are not known to be the most expressive people, as this drawing in *The social guidebook to Norway* indicates. Still, sentiment is present in Norwegian texts, and our models identify this sentiment on a fine-grained level. Image courtesy of www.thesocialguidebook.no

Chapter 2

Background

We here present the broader field of Sentiment Analysis and Targeted Sentiment Analysis. From previous work we learn about tools that have been commonly used in this field, and we learn about Sentiment Analysis from document level to sentiment target level. We present previous work for Targeted Sentiment Analysis, from statistical methods to the newest neural architectures. After the presentation of Sentiment Analysis and Targeted Sentiment Analysis from the English language sphere, the second part of this chapter presents challenges, resources and methods that are relevant for lower-resourced languages.

On the term "Sentiment Analysis" Sentiment Analysis, sometimes also called *opinion mining*, is the field of study that analyzes people's opinions, sentiments, or emotions toward something, expressed in written text (Liu 2017). Liu presents both "Sentiment Analysis" and "Opinion mining" to have first appeared in 2003. In this thesis we use "sentiment" and "opinion" as synonyms, and refer to this research field as "Sentiment Analysis".

Targeted Sentiment Analysis Targeted Sentiment Analysis is the task of identifying the *targets* for each expressed opinion, and the *polarity*, positive or negative, towards these opinion targets (Zhang et al. 2016). The task is also described as *Open-domain, targeted Sentiment Analysis* (Mitchell et al. 2013) or Target-Based Sentiment Analysis (Li et al. 2019).
The first part of identifying the sentiment target is extracting its boundaries, *target extraction*. Since the sentiment target can be a sequence of several words, we need to get the boundaries right, the start and end of the sentiment target. The second part of the task is *polarity classification*, where the sentiment towards the target is classified as positive or negative.

2.1 Sentiment Analysis: Previous work

Previous work within Sentiment Analysis is presented somewhat chronologically, with earlier rule-based systems for document classification first, before we move towards both a more fine-grained approach, and also into newer systems based on neural networks.

2.1.1 Lexicon-based Sentiment Analysis

When thinking about how we express sentiment and opinion towards a matter in text, we could, for both English and Norwegian, intuitively think of adjectives describing an entity. We consider the phrase *"This is my bike"* to be neutral in terms of sentiment, while *"This is my wonderful bike"* expresses a positive sentiment towards *"bike"*. We consider the word "wonderful" to convey a strong and consistent positive sentiment towards the entity described. Following this thought one may curate a lexicon of sentiment-bearing words, for their sentiment and intensity. Such a sentiment lexicon can be queried with the words in a text, to determine the overall sentiment. One such lexicon is the *Affective Norms for English Words* (ANEW) (Bradley and Lang 1999). Several thousands of words are given a two-decimal score between 1 and 9 for three dimensions describing human emotion, labeled *valence*, *arousal* and *dominance*. Sentiment analysis based on such lexicons are typically faster than machine learning, while machine learning enables training more accurate models.

A more unsupervised way of extracting a sentiment lexicon is presented by Turney (2002) who counted words co-occurring with the words "excellent" or "poor" in an AltaVista web search. In the text to evaluate, he collected word pairs containing adjectives or adverbs, and scored these to calculate the overall sentiment of the text. His experiments yielded an average accuracy of 74%, classifying reviews as positive or negative.

2.1.2 Sentiment classification with statistical methods

Dave et al. (2003) and Pang et al. (2002) present Sentiment Analysis on product reviews using the machine-learning techniques Naive Bayes and Support Vector Machines. Word features are selected, words are substituted, and n-grams are created and smoothed. Their method achieved 82% accuracy for document-level sentiment classification using Support Vector Machines.

2.1.3 Aspect-based Sentiment Analysis

Aspect-based Sentiment Analysis (ABSA) has been a popular branch within Sentiment Analysis, where normally the entire dataset consists of reviews within one domain, for instance hotels, restaurants or laptops. The task is, on a sentence level, to detect what aspects of the product is described as positive or negative. The aspect categories are usually predefined. For a restaurant dataset, the categories may be: [food, service, price, ambience, misc]. There were SEMEVAL shared tasks for Aspect Based Sentiment Analysis both in 2014, 2015 and 2016 (Pontiki et al. 2014, 2015, 2016). The top performing contribution for SEMEVAL16, identifying only what aspect categories were mentioned in each sentence, had an F_1-score of 84% (Pontiki et al. 2016).

A drawback with the aspect categories is that they need to be pre-defined and annotated for in the training data. To be manageable, the

entire datasets need some unifying theme, for instance "restaurant reviews" or "laptop reviews". In our Norwegian dataset, there is no such unifying theme, no annotations for aspect categories, and we do not include ABSA in our experiments.

2.1.4 Methods for Targeted Sentiment Analysis

Targeted Sentiment Analysis aims at for each sentence to extract sentiment targets and classify the polarity towards them. This is a newer task that has been tackled first with statistical methods for sequence labeling, and later with neural methods.

Zhang et al. (2015) present the transition in best performing architecture for Targeted Sentiment Analysis from a merely statistical approach where Conditional Random Fields (CRF) has performed well, into a combination of a neural architecture with fully connected layers and a CRF layer on top. This combination increases their F_1-scores overall with 5% and more, over a pure CRF-based model. For the task of detecting sentiment targets and the polarity towards them, they obtain an F_1-score of 40% at best. In recent years, a common neural architecture for Targeted Sentiment Analysis has been LSTM. Both LSTMs and the new Transformer-based architectures are introduced in the following sections.

We follow the common approach to treat Targeted Sentiment Analysis as a sequence labeling task. In sequence labeling, each word in a sentence receives a label. This label identifies whether or not the word is part of the sequence(s) the models seeks to extract from the sentence. Named Entity Recognition (NER) is one well-established sequence labeling task. Performing Targeted Sentiment Analysis as a sequence labeling task allows us to learn from literature and use software developed for sequence labeling. This is not the only possible way to approach Targeted Sentiment Analysis. Hu et al. (2019) point out weaknesses to this approach, both the large search space for the right label combination, and the possibility of invalid label sequences. In our work we did not find these limitations to be an important hindrance. Compute times are manageable, and invalid label combinations did not occur in the predictions that were checked. We therefore use sequence labeling methods only, in our work.

2.2 Experiments with LSTM

Recurrent neural networks (RNN), and especially networks with Long short-term memory (LSTM), have performed well on sequence-labeling tasks. Lample et al. (2016) presented bidirectional LSTMs with a CRF inference layer, a method that became dominant in sequence labeling the following years (Ma and Hovy 2016). The word-level LSTM structures are able to represent the global sequence information, and a CRF layer captures dependencies between neighboring labels. This has enabled many neural sequence labeling models to reach state-of-the-art performance (Panchendrarajan and Amaresan 2018; Yang et al. 2018). Li et al. (2019)

show how LSTM-based architectures originally trained for NER can be retrained with data for Targeted Sentiment Analysis and in this way become very strong models for this task.
A basic RNN works sequentially and each cell takes as an input, not only the representation of a word, but also the inner state from the previous calculation in that cell. Parameters are learnable, to set how much to focus on previous state, and how much to focus on new input. However, RNNs are difficult to train due to the vanishing gradient problem (Pascanu et al. 2013). The LSTM architecture is an improvement over the basic RNN in that it adds learnable sigmoid functions that provide *input gates*, *output gates* and *forget gates*. This allows the LSTM more flexibility in the mix between previous states and the new input. This has shown to reduce the vanishing gradient problem (Hochreiter and Schmidhuber 1997). Using a bidirectional LSTM allows the system to capture dependencies from both preceding words and following words in the text. The sentences are processed first left to right, and then right to left, and the outputs of the two runs are concatenated. The CRF inference layer on top of the LSTM layers finds tag sequences with the highest probability, based on both preceding and following outputs from the bidirectional LSTM. This setup, a *biLSTM-CRF*, is used in all our LSTM-based experiments.

2.2.1 Pretrained word embeddings

As an input to our neural network, each word needs a numerical representation. In earlier statistical systems one could represent each word with a "one-hot" vector with length equal to vocabulary size, where each word in the vocabulary gets its own location on the vector. The location representing a given word, gets a "1", while the other locations on the vector get "0". This way, each word has its own unique representation. But this representation does not carry any other information about the word. For neural networks we prefer pretrained word embeddings as representations for the words in our texts. These are dense vectors with usually between 50 and 1000 dimensions, where each value in the vector is set by machine learning, based on the co-occurrence the word has with other words. The aim is that synonymous words should have similar word embeddings. Firth (1957) popularized *The Distributional Hypothesis*, which states that words occurring in the same contexts, tend to have similar meaning. With large amounts of text available from the Internet, especially for English, it is possible to observe a large amount of words used in many sentences. Various algorithms and algorithm families have been created to work its way through large corpora. For each word the algorithm reads, it adjusts the values in the vector representation of that word, based on the co-occurring words.

For these algorithms, one important design choice is to define "co-occurrence". We do this by defining a "context window" where all words inside this window are counted as co-occurring with the word in focus. It is most common to set the focus word in the middle of the context window.

With the common "Continuous Bag of Words"-approach, each word in the context window is treated equally, disregarding whether the context word occurs before or after the focus word, adjacent to the focus word or further away in the window. We describe the window size by how many words are included in either direction. In this way, with a context window with size two, the two preceding words and the two following words are defined as co-occurring with the word in focus. Levy and Goldberg (2014) show an example of how the result can vary, based on this design choice. In table 2.1 we see that the resulting five word embeddings most similar to the word embedding for "florida", are mostly geographical entities inside the state of Florida. With a context window of two, names of two other states occur among the five most similar words to "florida".

Focus word	**Context window: 5**	**Context window: 2**
florida	gainesville	fla
	fla	alabama
	jacksonville	gainesville
	tampa	tallahassee
	lauderdale	texas

Table 2.1: The five most similar words to "florida", according to word embeddings trained with context window of 2 and 5 (Levy and Goldberg 2014).

Concerning the various algorithms, or families of algorithms developed for creating word embeddings, we mention three systems for their importance:

- Word2Vec
- GloVe
- FastText

The Word2Vec algorithms (Mikolov et al. 2013b) trains the network to assign high probability to the focus word given a context word, if this word-context combination has been seen during training. Any word-context combination not seen during training, should be assigned a low probability.

GloVe: Global Vectors for Word Representation (Pennington et al. 2014). Here, all co-occurrences are registered in a matrix for the n most common words. Matrix factorization methods are employed for generating low-dimensional word representations. Rare co-occurrences are given less weight, since these tend to be noisy.

FastText (Bojanowski et al. 2017) builds the vector representation not only on co-occurrence with other words, but also includes a subword model trained on character n-grams within the word. This approach tends to work well also with words that had few occurrences in the training corpus. With a subword model, rare words can get support from similar words, in finding its place in embedding space. Words that are similar in how they are spelled, may also be similar in their meaning, for instance if

one of them is an inflection of the other.

Putting word embeddings to use No matter how they are trained, these word embeddings make up a language model that in itself can be used to look up word similarities. The models are trained to give similar vector representations to words that occur in a similar context. The distance between these vectors can be measured by cosine similarity. But the most interesting use for these pretrained language models is using their word embeddings as input for machine-learning systems that subsequently are trained on specific NLP tasks.

Pennington et al. (2014) show the results of Named Entity Recognition with CRF, using as word representations either a set of discrete features, Word2Vec-based word embeddings, or GloVe-based word embeddings. Both Word2Vec-based and GloVe-based word embeddings lifted performance by several percentage points above models using discrete features. GloVe performed slightly better than Word2Vec for their task, and provided a new state-of-the-art system for NER at that time.

Yang et al. (2018) explore the benefits from initializing the first layer of an LSTM with such pretrained word embeddings as compared to random initialization. They find the improvements from using pretrained word embeddings to be significant, with results for their task increasing from below 85% to above 90%. This is consistent with the findings of Ma and Hovy (2016). For sequence labeling, Word2Vec gave less improvement over random initialization than GloVe. Schmitt et al. (2018) perform several ABSA experiments with Word2Vec, GloVe, and fastText word embeddings and LSTM. Their best performing models all use fastText word embeddings. Based on these results and our own earlier experiments, we use fastText word embeddings in all our experiments with an LSTM-based architecture.

Contextual word embeddings Word embeddings like those generated with fastText contain only one representation for each word. An ambiguous word like "bank" has its representation formed by all occurrences of "bank" during training, independent on its usage as a noun or a verb. The contextual word embeddings ELMo (Embeddings from Language Models) changed this (Peters et al. 2018). ELMo is an LSTM-based system for training a language model that retains information of a word's context. When queried with a sentence containing the word "bank", the model is able to output different representations of "bank", depending on the context in the query sentence. ELMo made contextual word embeddings widely available, and ELMo started the tradition of naming such models and tools after Sesame Street's Muppets.

2.2.2 Attention heads

Attaching attention mechanisms on top of the LSTM layers became a successful contribution towards improving the LSTM-based architecture. The attention mechanism was employed to improve the performance of an LSTM-based network in "Neural Machine Translation by Jointly Learning to Align and Translate." Here, Bahdanau et al. (2015) show how translation of longer sentences is improved by the attention mechanism. See Figure 2.1 on the next page. This approach is elaborated further by Luong et al. (2015), who describe the attention layer as a variable-length alignment weight vector. During translation this vector takes as input all RNN output states from the source sentence, and also the current target state. The learnable alignment weights open for the states that carry relevant information in order to predict the next word in the translation, and reduce the influence from states that do not contribute to the right translation. Wang et al. (2016) applied similar attention mechanisms successfully to Sentiment Analysis, and Baziotis et al. (2017) used two bidirectional LSTMs with attention heads to be among the top three contributors at SEMEVAL-2016 Task 4: Sentiment Analysis in Twitter, subtasks A though D (Nakov et al. 2016). Attention mechanisms are at the core of the Transformers architecture, which is discussed further in the following senction.

2.3 Transformers

In Targeted Sentiment Analysis, as in many other NLP tasks, we need to capture long-range dependencies throughout the sentence. We saw in section 2.2.2 that attention heads improved the performance of LSTM networks to capture such long-range dependencies. A question then arose: What if we got rid of LSTM-layers altogether, and encoded the entire sentence representation with attention heads? Vaswani et al. (2017) proposed the Transformer architecture that shows that this is indeed possible, *a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output*. The transformer is an encoder-decoder with a stack of multi-head self-attention layers. In these layers, the input is a matrix representation of a sentence where one dimension represents each element in the sentence, and another dimension is the vector representation for that word. Through matrix multiplications, the representations of each of the other words in the sentence become part of the representation for each word. Each of the n parallel heads in a multi-head layer, provide an output that is concatenated before going through a feed-forward network that provides the output of that multi-head self-attention layer. The attention mechanism described in the original paper, is called "Scaled dot-product attention"

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

The core of the multi-head attention-layers is illustrated in the original

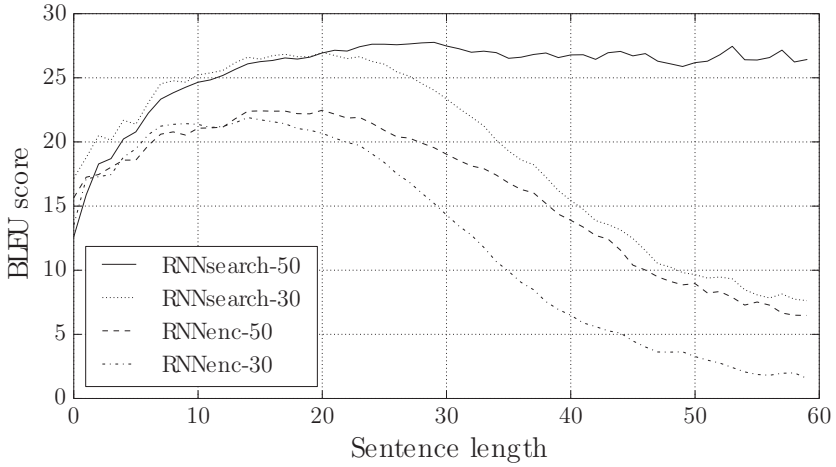


Figure 2.1: Translation quality with respect to the outputted sequence length: Translation quality increased for longer sentences with attention heads.

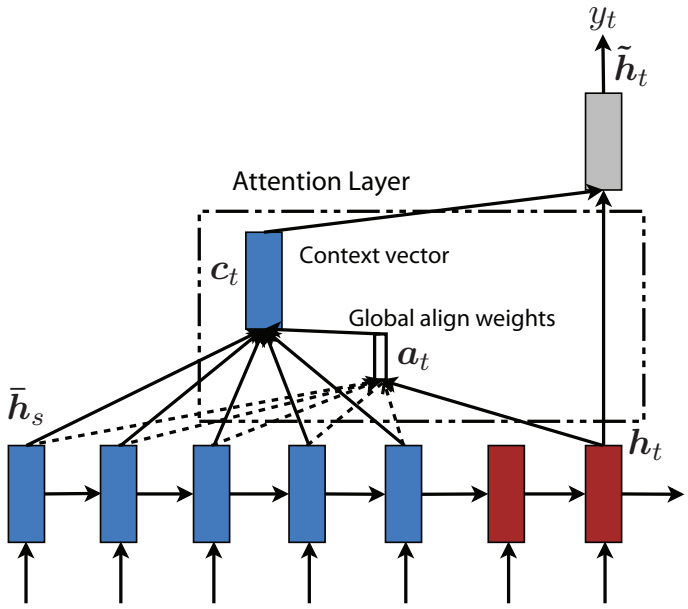


Figure 2.2: Attention mechanism during sequence-to-sequence neural machine translation (Luong et al. 2015): The attention head creates a context vector based on all hidden states from the input sentence, and the current word in the output sentence.

paper with figure 2.3, while the whole encoder-decoder is illustrated in figure 2.4 on the following page.

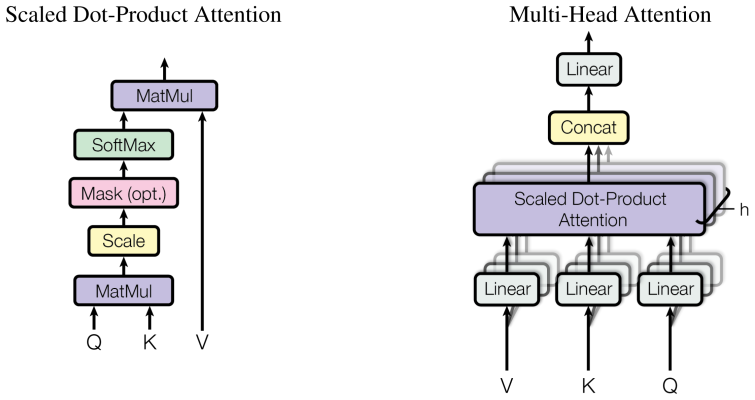


Figure 2.3: Left: Scaled dot-product attention. Right: Output from each attention head is concatenated and fed through a linear feed-forward network which provides the final output of the multi-head attention layer.

2.3.1 The BERT model

Based on the encoder part of the Transformers architecture, Devlin et al. (2019) developed the Bidirectional Encoder Representations from Transformers (BERT) pretrained language model. BERT is a bidirectional model with attention both to preceding and following words. One popular alternative to BERT, the GPT-architecture reads only from start to end in a sentence and is excelling in the more limited field of text generation. While there are other architectures to choose from, BERT is well documented and has been used in other research relevant to ours. There are versions of BERT implemented both with monolingual English data, and also with multilingual data that includes Norwegian. We therefore focus on BERT models in this thesis.

BERT training corpus and architecture The original $\text{BERT}_{\text{BASE}}$ model was trained on the BookCorpus, together with the English Wikipedia. The model has 12 attention heads in each layer, and 12 such Transformer layers. Maximum sequence length is 512 WordPiece tokens.

BERT preprocessing The input words have their accent markers removed, and whitespace is added around punctuation markers. The resulting space-separated tokens are split using WordPiece, a word segmenting algorithm by Wu et al. (2016).

BERT pre-training During pretraining, the model is trained on two tasks: Masked language modeling (MLM) and Next senctence prediction (NSP). In MLM, the model masks 15% of the words, and trains to predict the masked word. For NSP, the model is presented with two sentences from

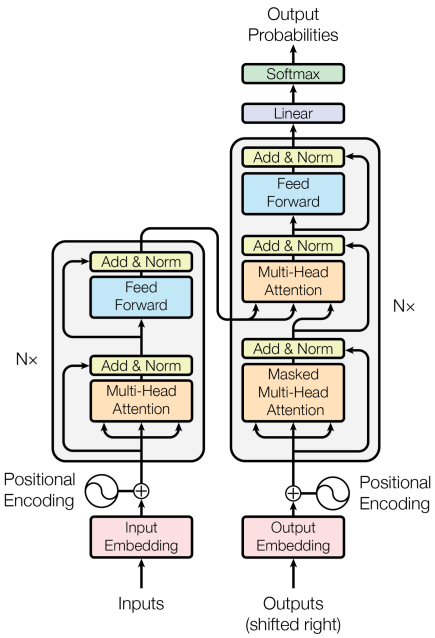


Figure 2.4: Multi-head attention layers are stacked to make up the encoder and decoder parts of the Transformer.

the corpus, and trains to predict if they follow each other or not, in their original setting.

2.3.2 Fine-tuning BERT for Targeted Sentiment Analysis

After the model has been pretrained, it is a fairly simple task to fine-tune the model for various downstream tasks. For sequence labeling, a fully connected layer is added on top of the BERT model, and trained to output the correct label based on BERT's representation of the words. During fine-tuning, the parameters of the final fully connected layer are learned from scratch, while the parameters of the BERT model itself are fine-tuned. Hu et al. (2019) show how sequence labeling with BERT provides new, strong baselines for Targeted Sentiment Analysis. This method achieves F_1-scores four to eight percentage points above the previous LSTM-based state-of-the-art models.

2.3.3 Advantages of Transformer-based architecture

A Transformer-based language model like BERT outputs contextual word embeddings, allowing the representation of a word to vary with different use cases. Both BERT and fastText models are pretrained on large amounts of text and has therefore seen common words in many contexts. But where a fastText model has only one representation for an ambiguous word like "bank", a BERT model output different representations of the word "bank" depending on the context, the sentence we use to query the model for word representations. In the following we comment on three other advantages

of this architecture: Parallel processing, easy adaptation to various tasks, and easier hyperparameter tuning:

Parallel processing We have noted that the LSTM architecture needs to process data sequentially. The output from processing the previous word serves as input to the process of the given word. A bidirectional LSTM needs to make another run in the opposite direction as well. These sequential processes can not be parallelized, and this is limiting on how efficient the system can be. Transformer-based models though, can be pretrained in parallel. They can therefore utilize recent advances in parallel processing power possessed by graphics processing units (GPU) and tensor processing units (TPU).

Task adaptation Although computationally costly to train, pretrained Transformer-based models like BERT can be fine-tuned for various tasks quite easily. Transformer-based models seem to work better for cross-domain tasks than LSTM-based models (Rietzler et al. 2020).

Hyperparameter tuning In our experiments, finding good hyperparameters was easy when fine-tuning a BERT-based model, and hard when training an LSTM-based model. We were able to go from good to better after a dozen experiments with hyperparameters for fine-tuning a BERT-based model. In contrast, we needed several hundred experiments to tune hyperparameters for the LSTM-based model.

2.3.4 Training cost and environmental considerations

Transformer-based language models are resource demanding to pretrain. There is a high amount of parameters to parameters involved, 110 million for $\text{BERT}_{\text{BASE}}$. Devlin et al. (2019) describe the computing power needed to train the BERT base and large models: *Training of $BERT_{BASE}$ was performed on 4 Cloud TPUs in Pod configuration (16 TPU chips total). Training of $BERT_{LARGE}$ was performed on 16 Cloud TPUs (64 TPU chips total). Each pre-training took 4 days to complete.* The environmental cost of training Transformer-based models is described by Strubell et al. (2019). They estimate that developing and training a large Transformer-based model with neural architecture search, represents a CO_2-emission of 626,155 lbs. For comparison, an average car during its lifetime, including production and fuel consumption, is estimated to represent a CO_2-emission of 126,000 lbs. Fortunately, there are many pretrained models available, and fine-tuning the models can be done at a very moderate cost. Although expensive to train, these models are downloaded and used in a large number of projects, simplifying NLP and advancing many research and production tasks. We therefore consider the energy, time and finances spent on developing these models, to be a worthwhile and valuable contribution.

2.3.5 Sentiment Analysis using Transformers

The most important reason to use Transformer-based models is because of its superior performance over LSTM in many NLP situations. In 2019 and 2020, we see that Transformer-based systems achieve new state-of-the-art results for many NLP tasks, including Sentiment Analysis. Ambartsoumian and Popowich (2018) find that Transformer-based models and other models with self-attention architectures outperform LSTM-based models for six different sentiment analysis tasks.

Rietzler et al. (2020) present a system for Targeted Sentiment Analysis based on $\text{BERT}_{\text{BASE}}$. They extract polarity given sentence and target for the restaurants dataset of SemEval 2014 and get an absolute improvement in accuracy of 2.2% over the previous state-of-the-art method. See also Arkhipov et al. (2019) and Sun et al. (2019a).

Cross-domain ABSA with BERT

BERT-based models have contributed to state-of-the-art performance on Aspect-Based Sentiment Analysis (Sun et al. 2019b). Rietzler et al. (2020) explore in their paper the cross-domain qualities of the BERT model. They use the SemEval 2014 Task 4 Subtask 2 datasets for restaurant and laptop reviews, and find that when fine-tuning their BERT-based model on one domain and tested on the other, their results were impressive. Also, fine-tuning with training data for both domains helps, although the testing is done on only one domain. In other words, one may use training data from one category to improve Targeted Sentiment Analysis for another category.

2.4 Low-resource languages

In this and the following sections, we present the situation for languages other than English, and present bilingual and multilingual tools that are helpful for Sentiment Analysis in a lower resourced scenario.

Norwegian is a small language with 5.3 million native speakers. It lacks the massive access to language data and resources that English has, but is privileged with a presence both on Google Translate and in the multilingual BERT language model, together with 100-110 other languages out of the more than 7100 languages actively used on the world today. The 104 languages present in the new multilingual BERT language model, make up less than 1.5% of the languages spoken in the world. However, these are the largest languages, and 69% of the world's population are native speakers of one of these 104 languages (Eberhard et al. 2019).

The Ethnologue Global dataset 2017 shows that half of the languages of the world do not have a written form. Although many of these are threatened, there are 1600 languages without a written form that still are in vigorous use. From the lowest resourced languages and up to the highest resourced languages, there are all levels of digital resources being available, like bilingual dictionaries, text corpora and annotated datasets. In their article "Multilingual Projection for Parsing Truly Low-Resource

Languages", Agić et al. (2016) present the task of NLP for low-resource languages this way:

> State-of-the-art approaches to inducing part-of speech (POS) taggers and dependency parsers only scale to a small fraction of the world's 6,900 languages. The major bottleneck is the lack of manually annotated resources for the vast majority of these languages, including languages spoken by millions, such as Marathi (73m), Hausa (50m), and Kurdish (30m).

Since low-resource languages lack the data needed to accomplish many NLP tasks, they risk missing out on the digital arena. Tools like machine translations, speech recognition and spell check use various NLP technologies. Finding ways to enable NLP for low-resource languages is therefore important, in order to support our rich language diversity into the future.

2.4.1 Some NLP resources for the Norwegian Language

We have mentioned that for Natural Language Processing, Norwegian has more resources than many other low-resource languages, with its presence on Google translate and in multilingual BERT. Members of the Language Technology Group[1] at the University of Oslo have released, or contributed towards some NLP datasets:

- Norwegian contribution in the Universal Dependencies project[2]
- For document-level Sentiment Analysis, Norwegian has NoReC: The Norwegian Review Corpus by Velldal et al. (2018) with more than 35,000 reviews. The reviews are labeled from 1 to 6, indicating the reviewer's sentiment towards what is reviewed.
- The NorNE corpus of named entities (Jørgensen et al. 2020)
- The $\text{NoReC}_{\text{fine}}$ dataset for fine-grained Sentiment Analysis, (Øvrelid et al. 2020) that is the main data source for our experiments, and is presented further in section 3.1 on page 23.

2.5 Cross-Lingual Sentiment Analysis

As for other NLP tasks, few languages have good corpora annotated for sentiment on the fine-grained level. Researchers have therefore sought to find methods for utilizing resources from a higher resourced language like English for Sentiment Analysis in a lower resourced target language. This gives us Cross-Lingual Sentiment Analysis (CLSA). Cross-lingual word embeddings are essential to much CLSA, and are presented later in this section, after examples of CLSA through other methods.

[1] https://www.mn.uio.no/ifi/english/research/groups/ltg/
[2] https://universaldependencies.org/

Machine-translated sentiment lexicon Mihalcea et al. (2007) present two early approaches for Cross-Lingual Sentiment Analysis. One method was machine translating an English sentiment lexicon to Romanian, their target language. This machine-translated Romanian sentiment lexicon was used as input to rule-based sentiment classification of Romanian text.

Transferring sentence-level sentiment annotations They also had a sentence-aligned Romanian-English corpus available, and applied a model for sentiment classification on the English side of the bilingual corpus. The sentiment classification of the English sentences was transferred to the Romanian sentences. These Romanian sentences with their sentiment labels became training data for a machine-learning model for Romanian sentence classification on the sentence level. Their main findings was that only a fraction of the words in the sentiment lexicon preserved their subjectivity during translation. To preserve subjectivity, corpus projections were found to be more reliable than lexicon translations.

2.6 Cross-lingual word embeddings

Aligning vectors from two or more languages is a core component for neural systems for Cross-Lingual Sentiment Analysis. Aligned bilingual or multilingual word embeddings open up for neural architectures similar to those used in monolingual Sentiment analysis. In this section we present the concept of cross-lingual word embeddings and how they may be helpful in transferring resources from one language to another.

2.6.1 Aligning vectors for two languages

When training word embeddings for the English language, the English word "house" will receive a representation close to the word "residence". The Norwegian word for "house" is "hus". If training a Norwegian model based on Norwegian text, the Norwegian word "hus" receives a vector representation unrelated to the representation of "house" in the English model. But the two models can be aligned through linear transformation so that "hus" becomes the nearest Norwegian word to "house". The linear transformation is learned by a loss function, originally least square error (Mikolov et al. 2013a). The loss is calculated on pivot pairs, pairs of words that are translations of each other. Several thousands of these pairs may be used in this process. Using least square error as loss function has some negative implications, one of which is the "hubness problem"(Dinu and Baroni 2015), that some word embeddings tend to become nearest neighbor of abnormally many other words. Conneau et al. (2018) introduced "Cross-domain Similarity Local Scaling" (CSLS) to compensate for the hubness problem. CSLS is developed further by Joulin et al. (2018) who obtain state-of-the-art performance with their aligning algorithm together with fastText pretrained language models. They have released word embeddings for 44 languages, each aligned with English. Norwegian is one of the languages

available, and these are the Norwegian-English CLWE in use for our Norwegian experiments involving LSTM[3].

Aligned word embeddings, Cross-Lingual Word Embeddings (CLWE) may be used for translation between the two languages. CLWE are also a tool for transferring knowledge between these two languages pairs, such that a neural NLP model trained for a task using labeled data from one of the two languages, can perform inference on data from the other language (Schuster et al. 2019). The described methods for aligning word embeddings require comprehensive text corpora for both languages, and a good dictionary for finding pivot pairs. For low-resource languages, these resources may not be available. In the following we present alternative methods for creating CLWE when one of the languages has a small corpus to train on, or where a good dictionary for finding pivot pairs is not available.

2.6.2 Alternatives for bilingual alignment

Different resource situations allow for different methods of creating and aligning cross-lingual word embeddings. Hermann and Blunsom (2014) use sentence-level aligned texts to align the word embeddings, while Vulić and Moens (2015) use document-level aligned texts instead of a bilingual lexicon. Upadhyay et al. (2016) present several such implementations of these approaches in *"Cross-lingual Models of Word Embeddings: An Empirical Comparison"*. Ziser and Reichart (2018) report good results training word embeddings both cross-lingually and cross-domain (CLCD). using a very low number of pivot pairs for alignment. Artetxe et al. (2017) align monolingual word embeddings using a reduced set of pivot pairs, from 5000 to 25. They also show that using numerals only as alignment pairs can work. With a low number of pivot pairs, the performance dropped remarkably when aligning Finnish and English, as compared to aligning the linguistically closer languages German and Italian.

Abdalla and Hirst (2017) show an integrated model for vector alignment and Cross-Lingual Sentiment Analysis using only 2000 pivot pairs. They used the ANEW annotations for sentiment values for individual English words, and assigned their sentiment values to their neighbors in the target language, after the vector space for the two languages was aligned. Even though translation accuracy for the whole vocabulary was quite poor with few pivot pairs, the sentiment classifier still performed relatively well.

When one language has little training data, a higher-resourced language can lift the performance of neural network language models for languages with little training data (Adams et al. 2017). CLWE were compared with monolingual word embeddings for scaled-down corpora on the target language side. CLWE performed better in all experiments. However, for sentence counts of 50,000 and above in the smaller training corpus, the difference was small.

[3] https://fasttext.cc/docs/en/aligned-vectors.html

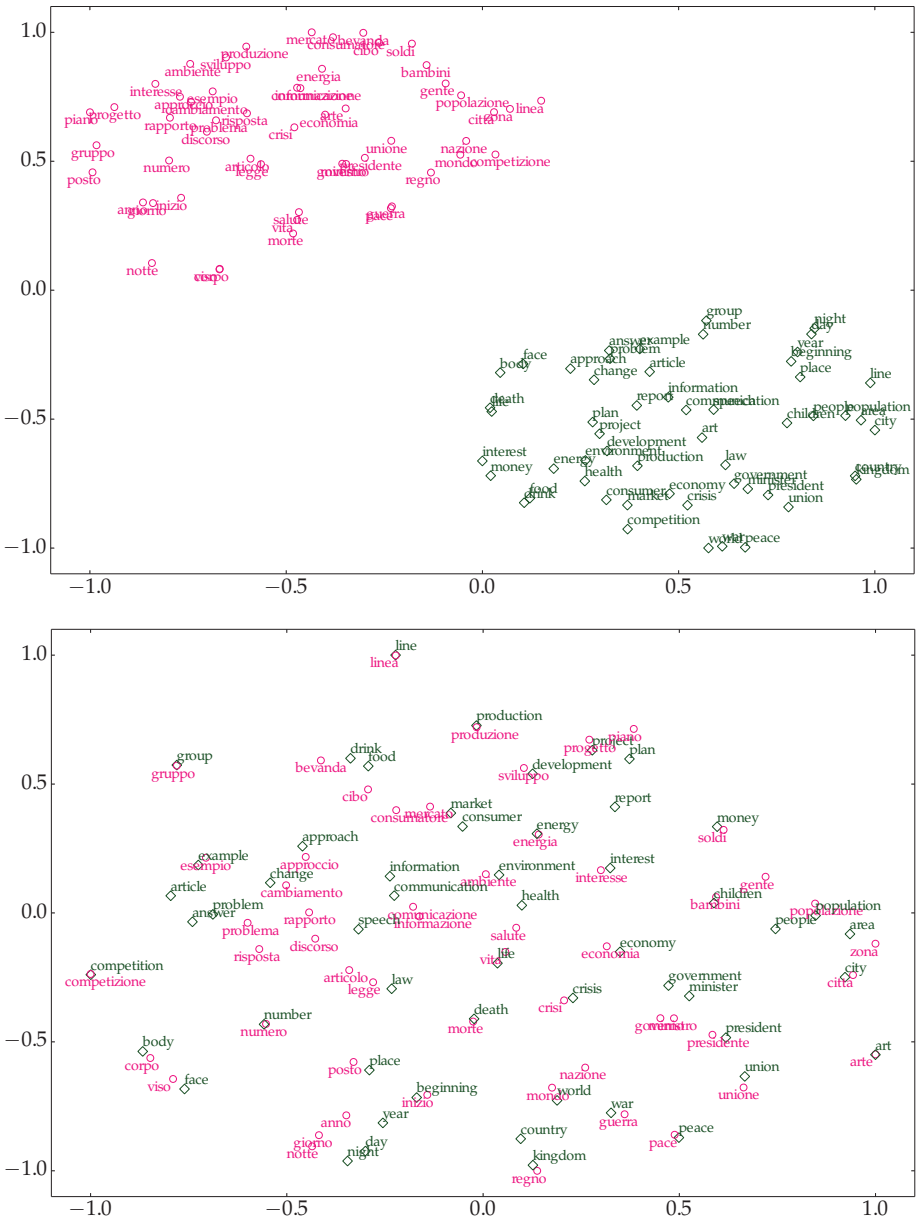


Figure 2.5: Aligning cross-lingual word embeddings: Moving from separate models for the two languages (top), to one model where similar words in both languages are close to each other (bottom). (Ruder et al. 2017)

2.7 Multilingual BERT

Multilingual BERT (M-BERT), released by Devlin et al. (2019), is a language model that was originally pretrained on the 100 languages with the largest Wikipedias. Additional languages have been added in more recent releases[4]. No special effort was made to align the languages, and language identification was not attached to the training sentences. The model architecture is similar to the monolingual BERT as presented in section 2.3.1 on page 13. The one major change is that the WordPiece vocabulary is increased from 30,000 to 110,000, to allow for the words from the other languages to be represented.

2.7.1 Multilingual performance of M-BERT

Although nothing is done to align the word representations from the different languages, experiments show that the languages are impressively well aligned and suited for cross-lingual tasks. It appears that M-BERT gives us "for free" much of the cross-lingual benefits that were sought after with CLWE. Pires et al. (2019) analyze the cross-lingual performance of M-BERT and show that M-BERT performs well cross-lingually, even when there is no lexical overlap, meaning that no words are written the same way in the two languages. An M-BERT model was fine-tuned for POS-tagging using only POS-labeled Urdu, written in Arabic script. This model achieved 91% accuracy on Hindi, written in Devanagari script. Although Urdu and Hindi are written with different scripts, spoken Urdu and Hindi are mutually intelligible as spoken languages, and may be considered two forms of the same language (Taj 1997).
For other language pairs the results were less encouraging. Different typological features serve as an explaination. With *typological features* we think of sentence segments orders like subject/object/verb order, or adjective/noun order. With few common features like these, the multilingual performance of M-BERT fell considerably. Karthikeyan et al. (2020) explored further the importance of lexical overlap, and confirm that lexical overlap contributes little to the cross-lingual abilities of M-BERT. The research of Karthikeyan et al. (2020) also supports the thought that cross-lingual performance of M-BERT correlates with similarity in typological features, and is not dependant on how many similar words there are in the two languages. Since Norwegian and English share many typological features, M-BERT may perform well cross-lingually between English and Norwegian.

Few-shot vs zero-shot

The multilingual performance of M-BERT has been evaluated through zero-shot experiments where M-BERT is fine-tuned for a task in one language and evaluated on the same task in another language (Karthikeyan

[4] https://github.com/google-research/bert

et al. 2020; Pires et al. 2019; Wu and Dredze 2019). M-BERT demonstrates in these experiments strong cross-lingual performance without any cross-lingual signal. M-BERT outperformed CLWE in four out of five NLP tasks in the experiments of Wu and Dredze (2019). They suggest as further work to add a small amount of target language supervision in these experiments.

When there is a need for Targeted Sentiment Analysis in a language covered by the M-BERT model, there is likely access to opinionated text in the target language. We assume also that it would be possible to annotate a few dozens of sentences for sentiment targets and the polarity towards them. This is our motivation for quantifying how well the M-BERT model can assist the task of Targeted Sentiment Analysis when there is some, but not much training data in the target language. If this is successful, none of the following requirements for performing Targeted Sentiment Analysis in a new language would be prohibitive.

a) A pretrained multilingual language model like M-BERT

b) Pre-existing training data from a higher resourced language

c) A manageable annotation process spanning hours or days instead of weeks or months

d) Moderate computing resources, using up to an hour on a single GPU instance for one experiment.

We contribute with such few-shot experiments in chapter 6. To our knowledge, these experiments provide new insights into the multilingual performance of M-BERT, beyond the zero-shot experiments found in existing literature.

Chapter 3

The datasets

This chapter presents the NoReC$_{\text{fine}}$ dataset. This dataset is the source of annotated data for Targeted Sentiment Analysis in Norwegian. We also present the English SEMEVAL14 Restaurants dataset which is our source of English data annotated for Targeted Sentiment Analysis.

3.1 NoReC$_{\text{fine}}$

The Norwegian Review Corpus NoReC (Velldal et al. 2018) is a collection of newspaper reviews regarding concerts, products, screen productions etc. A subset of this has been annotated for fine-grained sentiment, and is named NoReC$_{\text{fine}}$ (Øvrelid et al. 2020). The texts are annotated for polar expressions and their relation to a target and a holder. The relation to the target contains polarity and intensity. An overview of the entities in the annotation scheme and their relations is presented in Figure 3.1 on the next page.

In Table 3.2 on page 27, we see that there are 8634 sentences from 327 reviews in the training set. Two thirds of the targets receive positive sentiment, while one third receive negative sentiment. The words in NoReC$_{\text{fine}}$ are not lowercased or normalized. The dataset consists of 192,007 tokens, of which 30,305 are unique.

3.1.1 Domain diversity

The variety of review domains covered within NoReC$_{\text{fine}}$ sets this dataset apart from most other datasets for fine-grained Sentiment Analysis. Each review in the dataset belongs to a category, or domain, as presented in table 3.1 on the next page. "Screen" (Movies and TV-productions) and "Music" make up one third of the dataset each.

In contrast, all reviews in the SEMEVAL dataset presented later belong to the "Restaurants" domain. With reviews from a single domain only, the task of Sentiment Analysis is simpler. Multiple domains lead to more lexical variety, as shown in table 3.1. Also, with multiple domains come conflicting polarities expressed through adjectives like "broad", "heavy"

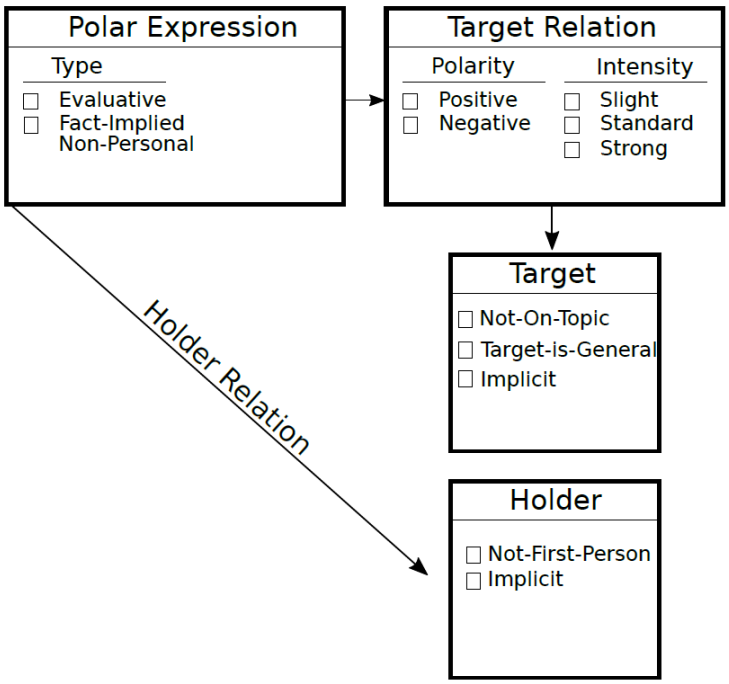


Figure 3.1: NoReC$_{\text{fine}}$ is annotated for polar expressions with relations to target and holder.

	Train		Total	
Categories	**sentences**	**documents**	**sentences**	**documents**
screen	2920	118	3806	149
music	1915	111	2692	144
products	1753	30	2181	39
literature	877	35	1089	42
games	445	16	767	23
restaurants	290	6	340	7
stage	249	8	376	11
sports	149	2	149	2
misc	36	1	36	1
Total	**8634**	**327**	**11436**	**418**

Table 3.1: Documents and sentences in the NoReC$_{\text{fine}}$ training and complete set by their domains.

or "slow". The adjective "slow" may be positive regarding cooking, but negative regarding laptops. The domain diversity of NoReC$_{\text{fine}}$ adds to the difficulty of Targeted Sentiment Analysis with this dataset. However, the domain diversity is also an asset, and opens for cross-domain experiments. There are more sentences in the "screen" category in NoReC$_{\text{fine}}$, that in the SEMEVAL14 Restaurants training set. The dataset is therefore suitable for experiments where models are fine-tuned with data from one domain, and evaluated on data from another domain.

3.1.2 Extracting sentiment targets from the annotations

The $\text{NoReC}_{\text{fine}}$ dataset comes with a script that aids extracting the sentiment targets with polarity, and label each word in the text accordingly. In most cases, this is straight forward based on the raw data in $\text{NoReC}_{\text{fine}}$: For each polar expression, read the target span and polarity from the annotations and tag the words in the target according to its polarity. However, a special case is when there are conflicting sentiments towards the same target. The annotations do not conclude about the winning polarity conveyed in the text towards these target expressions. When deciding what polarity to assign to this target, other datasets introduce the polarity category "conflicting". We do not use a "conflicting" category, and need to settle these conflicts as either a positive or negative polarity. Since polarities have intensity, we could let the strongest intensity win. We could count positive or negative expressions towards a target, or we could let the last polarity win. There is one sentiment target in each of the two sentences in example 2. The targets have conflicting opinion expressions towards them. For both cases, we consider it reasonable to let the last expression decide the overall polarity towards the target. The datasets we use have been converted using this rule for settling conflicting sentiments towards the same target.

Settling conflicting expressions towards a target There is one target in each of the two example sentences, with blue highlighting. Positive opinion expressions have green highlighting, negative expressions in red:

(2) Three positive expressions towards the target, standard intensity, and one negative expression with strong intensity:
Veronica Maggio er ei jordnær , søt jente som synger pent , men utstråler overraskende lite entusiasme og sjarm .

One negative expression with standard intensity, and one positive expression with slight intensity:
Det hele utarter seg bare til å bli en forutsigbar , men fin nok fremføring .

3.2 SEMEVAL Restaurants

The SemEval 2014 Restaurants dataset is our English language reference dataset (Pontiki et al. 2014). The texts are from user-submitted restaurant reviews, and the annotations include sentiment target and polarity. We obtained our copy of the dataset from The Sant project (Sentiment Analysis for Norwegian Text[1]). The dataset has 3843 sentences and 59,780 tokens, of which 6266 are unique. It is referred to as *SEMEVAL*, or *SEMEVAL restaurants* in this thesis.

[1] https://www.mn.uio.no/ifi/english/research/projects/sant/

3.3 A comparision between the datasets

Table 3.2 on the facing page shows that NoReC$_{\text{fine}}$ is a larger dataset than SEMEVAL, and has a considerably larger vocabulary. NoReC$_{\text{fine}}$ has positive sentiment towards two thirds of its targets, which is a bit more balanced than SEMEVAL.

Longer targets Figure 3.2 shows the distribution of sentiment target lengths in NoReC$_{\text{fine}}$ and SEMEVAL. NoReC$_{\text{fine}}$ has considerably more of the longer targets than SEMEVAL has. The SEMEVAL dataset is based on user reviews, while the NoReC$_{\text{fine}}$ is based on writings by professionals, mainly for printing in newspaper or the newspaper's website. We believe that this difference, together with different annotations instructions, account for the different target lengths. Since the entire target sequence needs to be correct for our evaluation to accept the predicted target, this task becomes harder for NoReC$_{\text{fine}}$. The larger variety of words also adds to the difficulty of our task of finding the sentiment targets and the polarity towards them.

Larger vocabulary The texts in NoReC$_{\text{fine}}$ are longer, come from a variety of domains, and use a larger variety of words. The SEMEVAL restaurants dataset is domain specific, all texts are restaurant reviews. It has shorter sentiment targets, and there is a more limited variety of words in these target expressions. These differences makes NoReC$_{\text{fine}}$ a more complex dataset to work with. Figure 3.3 on page 28 shows the relationship between corpus size and vocabulary size for the training set and full dataset for the English SEMEVAL and Norwegian NoReC$_{\text{fine}}$. The red line is approximately fitted to the Norwegian data according to Heap's law (Heaps 1978). The English data has less than half the vocabulary of what would be expected if the English data had the same vocabulary diversity as NoReC$_{\text{fine}}$.

	NoReC$_{\text{fine}}$		SEMEVAL	
	train	total	train	total
Sentences	8634	11436	2740	3843
Unique words per sentence	2,9	2,6	1,8	1,6
Targets	5044	6656	3293	3844
Negative targets	1558	2026	734	992
Positive targets	3486	4630	1902	2852
Total words in targets	9915	13090	3676	5424
Unique words in targets	4615	5815	1072	1411
Unique words per target	0.92	0.87	0.33	0.37
Average target length	2.0	2.0	1.4	1.4
Word count	144,245	192,007	42,543	59,780
Unique words	25,052	30,305	5007	6266

Table 3.2: Counts for sentences, sentiment targets and words. Training set and total counts for the two datasets. NoReC$_{\text{fine}}$ is larger than SEMEVAL, has longer targets and more unique words per sentence. NoReC$_{\text{fine}}$ is a more complex dataset to analyze than SEMEVAL.

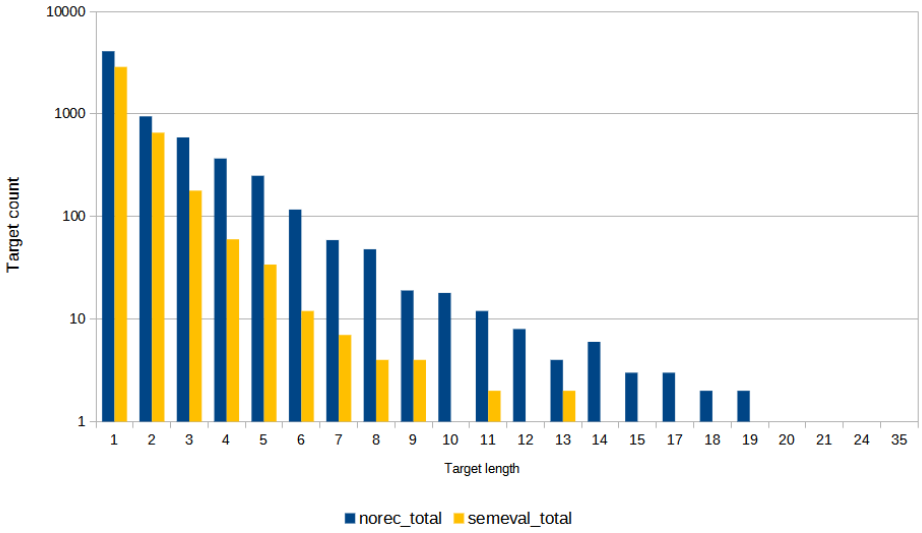


Figure 3.2: Total target counts in NoReC$_{\text{fine}}$ and SEMEVAL datasets, by target length.

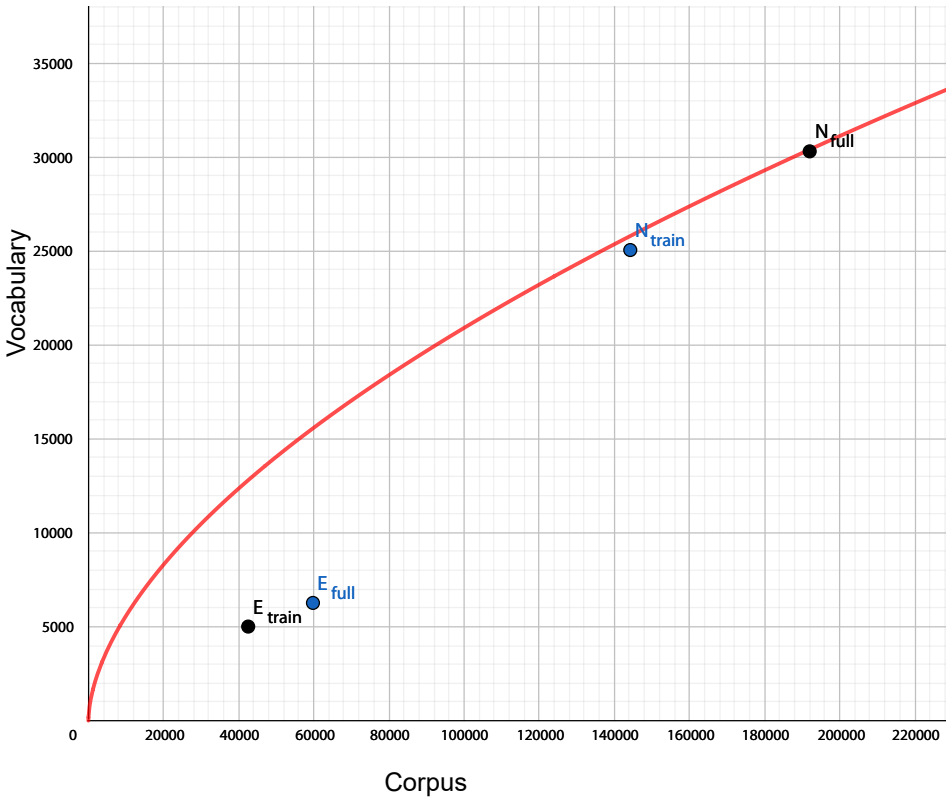


Figure 3.3: Vocabulary and corpus size for the Norwegian and English training data and full dataset. The red line indicates expected relationship between corpus and vocabulary for texts with the diversity of $\text{NoReC}_{\text{fine}}$.

Chapter 4

Monolingual experiments with NoReC$_{\text{fine}}$

In this chapter we seek to answer research question 1: *Which neural architecture is best for creating a model for Targeted Sentiment Analysis in Norwegian, based on the NoReC$_{fine}$ dataset*? We start with a pretrained language model, and fine-tune this with a suitable neural network for the task of Targeted Sentiment Analysis. As mentioned in section 2.2, we approach this as a sequence labeling task. We first present the technicalities around the experiments, then the experiments based on an LSTM architecture, before we present the experiments based on M-BERT. The results are compared and commented at the end of the chapter.

4.1 Experimental setup shared by all experiments

We created programs for data conversion to a unified CoNLL-U format, for scaling and mixing datasets, for iterating through hyperparameter settings, and for collecting and presenting the evaluations after each experiment.

Hardware for model training and fine-tuning

For this thesis, we were given access to the university's HPC cluster with GPU accelerations. However, most of our experiments are run on a laptop PC with a NVIDIA GTX 1070 GPU with the CUDA interface. Either way, training an LSTM-based model or fine-tuning a BERT-based model took 15-40 minutes, depending on the number of epochs and the size of the dataset.

Preprocessing

Traditionally in NLP, the words would be preprocessed to reduce the vocabulary needed to represent the text. In our datasets, the only preprocessing is tokenization. We use the term "token" to describe each basic character sequence used to represent the text in a useful way. In the tokenized, space-separated text "I love New York ." there are five tokens, the fifth being the period. No other preprocessing has taken place in our

datasets, neither by those preparing the dataset, or by us. The texts are not lowercased, and there is no lemmatization or removal of stop-words.

Since our texts include uppercased words, we need pretrained language models that are pretrained on texts including uppercased words as well. When lowercasing, the words "Angel" and "angel" become similar. This can be an advantage for rare words, since when a word is seen more often during training, its representation improves. On the other hand, when words are not rare, distinguishing "Angel" from "angel" retains information that allows for a more precise representation. As computing power and text source materials have increased, not lowercasing the texts seems to be the norm.

BIO tagging

We approach our task of identifying sentiment targets and polarity as a sequence labeling task similar to Named Entity Recognition (NER). Each word in the sentence is member of either no sequence, or one sequence with a category from a predefined list. When we extract sentiment targets with polarity, the sequence categories are "target-positive" or "target-negative", or "pos" and "neg" for simplicity. We adopt the BIO labeling scheme often used in NER, where each token is either outside (O) the entity we search for, or it is the first word (B) of an entity that we search for, or inside (I), meaning any other part of the entity, except the first word. Example 3 shows a sentence from a restaurant review, with the BIO tags for sentiment target and polarity.

(3)
```
O   B-NEG O   O   O      O   O   B-POS I-POS I-POS   I-POS
The food  can get pricey but the prixe fixe  tasting menu

  O  O   O
  is the greatest
```

CoNLL-U text file format

The texts and their labels are stored in the CoNLL-U format. This approach is the most commonly used in literature, and we find it effective for our purposes. It has one word on each line, followed by its tag. Word and tag are either separated by space or tab. The sentences are separated by a blank line.

Dataset with or without polarity in tags

Training models for sentiment target extraction only is easier than training a model for both target extraction and polarity classification. One approach to the two tasks could be to jointly extract targets and classify polarity,

and afterwards strip polarity information from the predicted tags. An other approach is to first extract targets only, and to feed this prediction to a second model for classifying polarity. In introductory experiments we found creating separate models for the two scenarios to be our best option. We train one model for the task of target extraction only, and another model for jointly extracting targets and classifying polarity. We got 3-4 percentage points better F_1-score with this approach, and we keep this approach through all experiments.

4.2 Evaluation metrics

When our model has predicted the sentiment targets in a text, we need to establish meaningful evaluation metrics. We present how this is done in our work, with a focus on the entity level.

Token-level accuracy

Each token in the texts in our experiments, has a tag according to the BIO scheme presented in section 4.1. We compute accuracy from counting the number of correctly assigned labels during prediction, divided by total number of tokens.

$$Accuracy = \frac{C_{corr}}{C_{total}} \tag{4.1}$$

Evaluation scheme for multi-token entities

Since our sentiment targets may span more than one word, it is of interest to inspect not only how many of the words were tagged correctly, but also how the annotated entities come out in the prediction.

Precision and recall For each category of tags, in our case *pos* and *neg*, the annotations provide a number of token sequences belonging to this category. We use Precision (P) and Recall (R) to measure the quality of our predictions for each category. Precision looks at the sequences we predicted and how many of them were truly sequences belonging to the predicted class. With recall we look at the token sequences in the annotations, and measure how many of these are found in the predictions. These relations are expressed in example 4.2 where TP = True Positives, FP = False Positives, and FN = False Negatives.

$$P = \frac{TP}{TP + FP} \qquad\qquad R = \frac{TP}{TP + FN} \tag{4.2}$$

A model that predicts too many sequences would have few false negatives, but many false positives. This would give a recall close to 1, and a precision close to 0. A model that returns only a handful of token sequences, those with highest probability of belonging to the category, would get very few false positives. Precision could be high, but recall

would be low. A metric that balances precision and recall, is F_1-score, which is the harmonic mean of the two.

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{4.3}$$

To find the average of the F_1-score for the entire testing set, we may count all the true and false positives and negatives for all classes combined. This way, the results for the largest category weigh more in the average. This is the micro-averaged F_1-score. In our work we report micro-averaged F_1-scores.

Strict or realxed evaluation

We mostly use a *strict evaluation* for which sequences are counted as correct. Here, the entire sequence needs to be correct, in order for the predicted entity to count. When we train for predicting polarity as well, the polarity also needs to be correct for the sequence to be counted. This strict definition equals the requirements for a successful prediction from CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition (Tjong Kim Sang and De Meulder 2003).

For comparison with previous work we use *relaxed evaluation* in some final tests. With relaxed evaluation, annotated and predicted sequences only need to have some overlap in order to be counted as correct, as long as polarity is correct. Many sentiment targets in $\text{NoReC}_{\text{fine}}$ are lengthy, and identifying the exact boundaries for such entities is hard, even for humans (Wiebe et al. 2005). Our relaxed evaluation follows *binary* evaluation metric presented by Katiyar and Cardie (2016), and is also used in the paper presenting $\text{NoReC}_{\text{fine}}$. Example evaluations for a sentence, using strict and relaxed evaluation schemes, are presented in Table 4.1 on the facing page. When not otherwise specified, we use the strict evaluation scheme for identifying successfully predicted sentiment target sequences, and report the micro-averaged F_1-score as default.

4.3 Setup for LSTM-based experiments

The bidirectional LSTM architecture with a CRF inference layer has provided state-of-the-art performance on Targeted Sentiment Analysis, and was on top of leaderboards at the beginning of this thesis work. We present here our biLSTM-CRF model of Norwegian Targeted Sentiment Analysis, trained and tested on $\text{NoReC}_{\text{fine}}$. We do these experiments both for the joint detection of target boundaries with polarity, and for detecting target boundaries only. We present here our chosen framework for these experiments and the work with word embeddings. In the next section we present the hyperparameter tuning and the results from these experiments.

Evaluating target boundaries and polarity			**Strict**		**Relaxed**	
Text	Gold	Predicted	Recall	Precision	Recall	Precision
The	O	O				
waitress	B-targ-pos	B-targ-neg	0	0	0	0
suggested	O	O				
glasses	B-targ-pos	O	0		1	
of	I-targ-pos	O				
wine	I-targ-pos	B-targ-pos		0		1
that	O	O				
went	O	O				
very	O	O				
well	O	O				
with	O	O				
the	O	O				
food	O	B-targ-pos		0		0
.	O	O				
			0	0	0.5	0.33

Evaluating target boundaries only			**Strict**		**Relaxed**	
Text	Gold	Predicted	Recall	Precision	Recall	Precision
The	O	O				
waitress	B-targ	B-targ	1	1	1	1
suggested	O	O				
glasses	B-targ	O	0		1	
of	I-targ	O				
wine	I-targ	B-targ		0		1
that	O	O				
went	O	O				
very	O	O				
well	O	O				
with	O	O				
the	O	O				
food	O	B-targ		0		0
.	O	O				
			0.5	0.33	1	0.67

Table 4.1: Example sentence with precision and recall for the two different evaluation schemes *strict* and *relaxed*.

4.3.1 Framework for LSTM-based experiments

We use NCRF++ for LSTM-based experiments, a unified neural sequence labeling framework to reproduce and compare recent state-of-the-art models with different configurations (Yang et al. 2018). The framework is available on github[1]. The framework facilitates rapid implementation of neural models with LSTM and CRF for sequence labeling. The system provides micro-averaged F_1-scores based on the strict entity evaluation scheme. We added to this framework code for hyperparameter search and for effective collection of results from multiple experiments.

4.3.2 Pretrained word embeddings

We let pretrained word embeddings represent each word in the datasets and initialize the first layer of the LSTM with these. These pretrained word embeddings have been trained on large corpora, and they have been given their values "from the company that they keep". We experimented with selected pretrained word embeddings among the various fastText pretrained models available through the NLPL word embeddings repository. For both the English and Norwegian monolingual experiments, we found that word embeddings from models trained on a larger corpus performed better than those trained on a smaller corpus. We also found that a vector size of 300 was better than 100, and that increasing the size beyond 300 did not give us any better performance. This was tested by running individual experiments for Targeted Sentiment Analysis, and checking performance on the dev set with a few selected pretrained models. For each pretrained model we tried altering a few hyperparameters, like dimensions of the hidden layers.

The researchers behind fastText have later released their own set of pretrained word embeddings that cover Norwegian[2]. We compared these models with the previously tested models, and found them equally good or better for our task. We tested the Norwegian and English monolingual vectors presented by Grave et al. (2018), and the aligned Norwegian-English vectors presented by Joulin et al. (2018). Table 4.2 on the next page shows the results from training models for Targeted Sentiment Analysis with the two alternative word embeddings, where the bilingual word embeddings performed better at both tasks. The observation that bilingual word embeddings perform better than monolingual, even on a monolingual task, is in line with the findings of Adams et al. (2017). There was a considerably higher out-of-vocabulary count when using the bilingual vectors, but since measured performance was higher, we chose to use these bilingual vectors from fastText for all LSTM experiments. Table 4.2 shows F_1-scores testing on the $\text{NoReC}_{\text{fine}}$ dev set with the two vector models.

[1] https://github.com/jiesutd/NCRFpp
[2] https://fasttext.cc/

Task	monolingual fastText no	bilingual fastText en-no
Target+pol	0.245	**0.271**
Target only	0.348	**0.374**

Table 4.2: F_1-scores for models with two alternative pretrained word embeddings. The bilingual word embeddings resulted in higher F_1-score, and we use these for all LSTM-based experiments.

4.4 LSTM experiments and results

We here describe our hyperparameter tuning, before reporting our best results for both tasks of detecting sentiment target boundaries only, and for detecting sentiment target and polarity.

4.4.1 Hyperparameter tuning

With the environment we have set up for our experiments, we were able to experiment with many combinations of hyperparameters. We performed an extensive grid search on the hyperparameters we considered to be most important, based on our previous experience with sequence labeling and LSTM. Some hyperparameters explored, with the max-min settings are:

- **Dimensions for the neural network:** Batch size (5-180), Hidden dimensions (150-250), Number of LSTM layers (1,2).
- **Settings for the network's learning:** Learning rate (0.007-0.2), optimizer (separate table), l2 regulation (1e-9, 1e-6), dropout (0.15, 0.23).

We searched through these and other settings for the best combination of hyperparameters, and chose those which gave the best F_1-score from one evaluation on the dev set. There was one exception from the rule of choosing the highest yielding F_1-score: L2 weight decay and dropout are known to improve generalization (Srivastava et al. 2014), and we therefore adjusted these hyperparameters slightly above what gave us the best result on the dev set. Weight decay was adjusted from 1e-9 to 1e-8, and dropout from 0.15 to 0.18. Further increasing these values had too much effect on the performance of the dev set. Selected hyperparameters are shown in table 4.3 on page 37. We found two hyperparameter settings that might be worth mentioning in our best setup: Batch size and choice of optimizer.

- We found a batch size of only 5 to perform better than larger batch sizes. At several times during hyperparameter search did we try to increase batch size, but found performance to drop each time. Since we could afford the time needed, we let batch size stay at 5.
- We found the SGD optimizer to work better than the Adagrad - ADAM-related optimizers. See Table 4.4 on page 37. This

corresponds with the finding of Yang et al. (2018), who also cite other sequence labeling experiments where SGD has been the best, and one paper where SGD performed the poorest.

4.4.2 Evaluation on the held-out test set

With the hyperparameters optimized on the dev set, we ran seven tests for each task on the held-out test set, in order to determine variance. We trained new models for each of these tests, where any locking of the random seeds was removed. The average F_1-scores are reported in table 4.5 on the facing page. We see a couple of percentage points lower performance in the test set, indicating a slight overfitting. Evaluation of the experiments are with strict scheme where all tokens in the sequence need to be correct, for the prediction to count. We also ran an evaluation on one model using the relaxed scheme where only one token overlap is needed for the prediction to count. The results are carried over to table 4.7 on page 39, for comparison with alternative models.

4.5 Setup for BERT-based experiments

We performed the same experiments with the BERT Transformer-based architecture, as we did with LSTM-based architecture. We used the multilingual M-BERT, introduced in section 2.7 for these experiments with Norwegian data. We were not aware of any pretrained monolingual Norwegian BERT-related model at the time of performing these experiments. M-BERT is the multilingual Transformer-based model with the longest history to our knowledge, and there are at least a few papers available, analyzing the multilingual capacities of this model, making it the model of choice for this work.

Terminology When doing experiments with LSTM-based models, we speak of *training* a model, based on *pre-trained word embeddings*. When doing experiments with Transformer-based models, we speak of *fine-tuning* a *pre-trained language model*. When speaking jointly of models of both kinds, we use *fine-tuning*.

Adapting BERT for sequence tagging To fine-tune a BERT-based model for Targeted Sentiment Analysis, a fully connected neural layer is added on top of the pretrained BERT model. The parameters of the final layer are set during fine-tuning on our training data. The weights of the BERT model itself are also fine-tuned during this process. For our experiments with BERT-based models, we used simpletransformers[3]. Simpletransformers provide an abstraction layer on top of the popular Python package *Transformers* by Huggingface (Wolf et al. 2019).

[3] https://simpletransformers.ai/

Selected hyperparameter settings	
optimizer=SGD	iteration=28
cnn_layer=2	batch_size=5
hidden_dim=180	char_hidden_dim=40
lstm_layer=1	bilstm=True
dropout=0.18	learning_rate=0.01
word_emb_dim=50	lr_decay=0.05
char_emb_dim=35	momentum=0
use_crf=True	l2=1e-8
use_char=True	gpu=True
word_seq_feature=LSTM	char_seq_feature=LSTM

Table 4.3: Hyperparameter settings for our best LSTM-based model for Targeted Sentiment Analysis on $\text{NoReC}_{\text{fine}}$

Performance with optimizer alternatives					
optimizer	momentum	Accuracy	Precision	Recall	F_1-score
ADAM		0.8529	0.3580	0.3333	0.3452
AdaDelta		0.8851	0.5341	0.2548	0.3450
Adagrad		0.8831	0.4867	0.3506	0.4076
RMSprop		0.8616	0.3702	0.3525	0.3611
SGD	0	0.8910	0.4839	0.4598	**0.4715**
SGD	0.1	**0.8917**	0.4810	0.4617	0.4712
SGD	0.2	0.8877	0.4656	0.4674	0.4665
SGD	0.3	0.8880	0.4944	0.4253	0.4573
SGD	0.4	0.8868	0.4956	0.4330	0.4622
SGD	0.5	0.8861	0.4577	0.4349	0.4460

Table 4.4: Checking alternative optimizers for the LSTM-based model when training on sentiment target boundaries only.

Task	# models	F-score	St dev
Targets with polarity	7	0.2357	1.80%
Target boundaries only	7	0.3383	1.95%

Table 4.5: F_1-scores for evaluating our best LSTM-based models trained on the $\text{NoReC}_{\text{fine}}$ training set and tested on the test set. We rebuilt 7 models and report the average F_1-score and standard deviation.

The system has a predefined pipeline for sequence tagging (NER), which works for our task without further adaptations.

4.6 BERT-based experiments and results

This section describes the hyperparameters we tuned during our experiments and the evaluation results from our best performing models.

4.6.1 Hyperparameter tuning

The default hyperparameters in the simpletransformers pipeline worked well with our data. We tested on all epoch counts from 3 to 20, and decided on 8 epochs. The increased performance after 8 epochs was minimal. Weight decay was the only other hyperparameter that was adjusted, which ended on 0.001. Batch size was kept at 32. These settings were not changed for any of the experiments reported here.

4.6.2 M-BERT Evaluation results

Training on the $\text{NoReC}_{\text{fine}}$ dataset and testing on the test set, we got the results shown in table 4.6. There was a drop in F_1-score between dev and test of between 1 and 1.5 percentage points. Standard deviation in the sets of testruns is lower here than with the LSTM-based models.

Task	# models	F-score	St dev
Targets with polarity	7	0.3889	1.07%
Target boundaries only	7	0.5105	0.52%

Table 4.6: Our best BERT-based model for Targeted Sentiment Analysis in Norwegian, tested on the $\text{NoReC}_{\text{fine}}$ final test set. Fine-tuning and evaluation repeated 7 times.

4.7 Best models for Norwegian Targeted Sentiment Analysis

The experiments in this chapter answer RQ 1: *Which neural architecture is best for creating a model for Targeted Sentiment Analysis in Norwegian, based on the NoReC_{fine} dataset?*
We now compare the LSTM-based architectures with our system based on M-BERT. LSTM-based models have been the first choice for tasks like ours for a few years, while BERT-based models have become increasingly popular the last year or two. We find that M-BERT outperforms LSTM for this task. Even though Norwegian is only one of more than a hundred languages represented in M-BERT, we were not able to create any LSTM-based model that came near M-BERT in performance. The previously

Targets with polarity, strict evaluation		
Architecture	**F-score**	**St dev**
LSTM	0.2357	1.80%
M-BERT	**0.3889**	1.07%

Target boundaries only, strict evaluation		
Architecture	**F-score**	**St dev**
LSTM	0.3383	1.95%
M-BERT	**0.5105**	0.52%

Target boundaries only, relaxed evaluation		
Architecture	**F-score**	**Origin**
LSTM	0.3910	Baseline
LSTM	0.4262	Our model
M-BERT	**0.5958**	Our model

Table 4.7: Best models for Targeted Sentiment Analysis on $\text{NoReC}_{\text{fine}}$. For both tasks and with both strict and relaxed evaluation, the model based on M-BERT performed 15 percentage points or more above best LSTM-based model.

reported results for LSTM and M-BERT are compared in table 4.7. The lower variance between models based on M-BERT is another advantage with this architecture.

Table 4.7 also shows that our LSTM-based model is slightly better than the baseline published by Øvrelid et al. (2020). We believe a reason for this is that the baseline experiment extracts sentiment target, expression and holder in one model. Since the annotations for sentiment target and sentiment expression are allowed to overlap, this demands some compromise to the labeling of tokens belonging to both target and expression. M-BERT confirms its superior performance when evaluated with the relaxed scheme. The $\text{NoReC}_{\text{fine}}$ dataset is reported to have an F_1-score of 73% for inter-annotator agreement for the sentiment targets, evaluated with the relaxed scheme. Our best model has 60% agreement with the provided annotations, using the same evaluation scheme. The results from these experiments are compared with alternative methods in the following chapter.

Chapter 5

Bilingual experiments

We found in chapter 4 that our best system for Targeted Sentiment Analysis on Norwegian text was based on fine-tuning the multilingual M-BERT with Norwegian training data. In this chapter we utilize resources from another language, English, in an attempt to improve this method. This will answer RQ2: *Are there any multilingual language resources that can improve our Norwegian model for Targeted Sentiment Analysis?* The first method to test is machine translation into English. When all data are in English, better monolingual English pretrained language models can be utilized. Some details are always lost in translation, but some performance may be gained from being able to use English-only tools. We present the process of machine-translating the dataset, and we present the chosen pretrained model for this task.

The second method explored in this chapter, is mixing English and Norwegian training data. We mentioned in section 2.7 that the language representations in M-BERT are impressively well aligned. When an M-BERT model is fine-tuned for a task in one language, it can perform well on that task in another language with similar typological features. In this chapter we use the Norwegian training data and add to that the SEMEVAL dataset. Evaluation is still on Norwegian testing data only. We do this experiments also with fastText bilingual word embeddings and LSTM. Rietzler et al. (2020) report that for Aspect-Based Sentiment Analysis, performance of their BERT-based model improved if fine-tuned on additional data from a different domain, together with training data from the domain they tested on. Our experiment attempts to improve a model by adding data that is from both another language and another domain.

The experiments in this chapters show that adding English training data matches, but does not surpass the performance of the model fine-tuned on Norwegian data only. The experiment with machine-translated data performs better than the monolingual LSTM-based model, but weaker than the monolingual M-BERT model from chapter 4. Both approaches seem to be relevant for other resource situations, although they did not improve performance in our case.

5.1 The datasets for bilingual experiments

For the experiments with fine-tuning on $\text{NoReC}_{\text{fine}}$ and SEMEVAL together, the sentences in the two training sets were shuffled randomly. We did preliminary experiments where we tried both appending the Norwegian data to the English, and shuffling the data randomly. We found no particular difference for the two approaches, and chose to always mix by random shuffling. Once the datasets were created, we used the same mixed datasets for all experiments. Testing is on the $\text{NoReC}_{\text{fine}}$ dev and test sets. The following parts of this section present the process of machine translating $\text{NoReC}_{\text{fine}}$ into English, and an attempt to quantify the errors introduced by this process.

5.1.1 $\text{NoReC}_{\text{fine}}$ machine-translated into English

We translated the $\text{NoReC}_{\text{fine}}$ texts with the google translate api, retokenized the text and transferred tags with *fastalign* (Dyer et al. 2013). We followed the guidelines for asymmetric alignments[1]. The fastalign software connects each word in the translated text with a word in the source text. This way, we can tag the English words with the same tag as the connected Norwegian word. With asymmetric alignment we run this process from source to target language only, without any attempt to align back from target to source. As the algotihm is noisy, we devised a post-processing scheme to defragment the opinion targets so that one contiguous target in Norwegian would be transferred to one contiguous target in English. As a result of this process, we have a machine translated English dataset with the texts from $\text{NoReC}_{\text{fine}}$, with tags for sentiment targets and polarity.

5.1.2 Manually evaluate 100 machine translated sentences

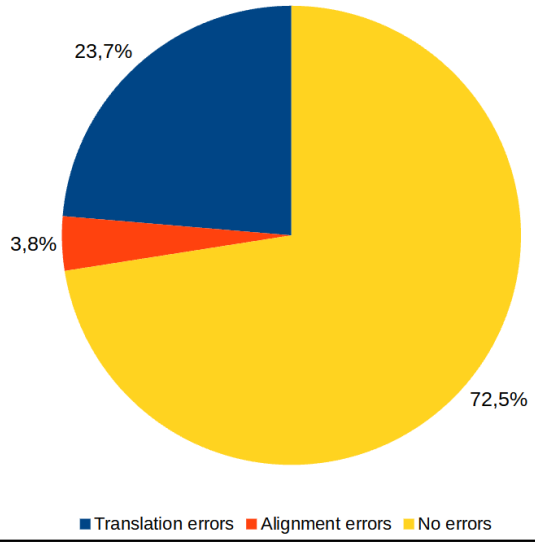
Both the machine translation and the transfer of tags are sources of error that degrades the dataset to some degree. In order to quantify these errors, we sampled randomly one hundred sentences from $\text{NoReC}_{\text{fine}}$ and inspected the Norwegian source text and the English translation. Observations from these samples indicate how well the sentiment targets were preserved through translation, and how well the opinion polarity was preserved.

We counted the total amount of targets in the 100 Norwegian sentences, and counted how many of these targets were present in the machine translated sentences, and whether the opinion towards the targets were the same in the machine translated texts as in the original.

- We first checked whether the original sentiment targets were present in the English translation, and whether the English sentence contained the same sentiment towards these targets as the Norwegian source.

[1] https://github.com/clab/fast_align

- For the targets successfully translated into English, we checked whether the targets were tagged correctly.



Outcome for target	Count	Percent
Translation error	31	23.70%
Alignments error	5	3.80%
No errors	95	72.50%
Total Norwegian targets	131	100.00%

Figure 5.1: In the 100 sentences we evaluated, there were 131 targets. We considered 72.5% of these to be preserved as sentiment targets through translation and tags transfer.

We found most of the errors in the inspected translations to be in the machine translation of the opinion expressions towards the target. The text representing the sentiment target entity was present, but the sentiment that the Norwegian text conveyed towards the target, was distorted in several occasions. It was difficult to set the limit for whether the sentiment towards the target was lost or not. However, we decided that for 23.7% of the targets, the sentiment expression towards them was lost. We also found in these 100 sentences several examples where we would disagree with the gold annotations. Figure 5.1 shows that after machine translation and transfer of tags, we considered 72.5% of the targets with opinion to be intact in the English translation.

This evaluation indicates how sensitive targeted Sentiment Analysis is towards the details of a language. Example 4shows one translation error that alters the sentiment towards the target. The Norwegian word *kostelige* is translated *expensive*, while a better translation would be *precious* or *priceless*. The target *" buddy cop " -scener* is annotated for positive sentiment, which is lost in the wrong translation of *kostelige*.

(4) **Norwegian original:**
Og mot slutten er det noen riktig så kostelige " buddy cop " -scener som gir godt med humør mellom de to politipartnerne .
English machine translation:
And towards the end , there are some really expensive buddy cop scenes that give a good mood between the two police partners .

Thoughts on the machine translated dataset We have created a machine translated version of $\text{NoReC}_{\text{fine}}$ that can be analyzed using monolingual English tools. We have looked briefly into the question about how much an analysis of the translated text is a valid analysis of the original Norwegian text. By inspecting one hundred sentences in Norwegian and English, we considered the targets, the sentiment towards them and the labeling in the English translation to correspond with the Norwegian original for 72.5% of the targets. By inspecting the sentences we were reminded of how hard it is in general to identify sentiment targets. This is reflected in the relatively low inter-annotator agreement of 73% for target identification in $\text{NoReC}_{\text{fine}}$. This uncertainty increases with the not-perfect machine translation. We consider this English machine-translated dataset to be an interesting, although not perfect representation of the Norwegian text, as long as one can work with the inference in English.

5.2 Experiments with Machine-translated text

The English machine-translated version of $\text{NoReC}_{\text{fine}}$ can be analyzed using monolingual English resources. Both Mihalcea et al. (2007) and Barnes and Klinger (2019) point out difficulties when using machine translation in Sentiment Analysis. Given the recent improvements in machine translation and the typological similarities between English and Norwegian, we found the method worth exploring.

We present here our choice of model, and experiments with the machine-tranlslated $\text{NoReC}_{\text{fine}}$. We use *MT* to indicate that the data are machine-translated from Norwegian to English. The Transformer-based models have performed notably better than the alternative so far. The following experiments are with Transformer-based pretrained models only.

5.2.1 Choosing English pretrained model

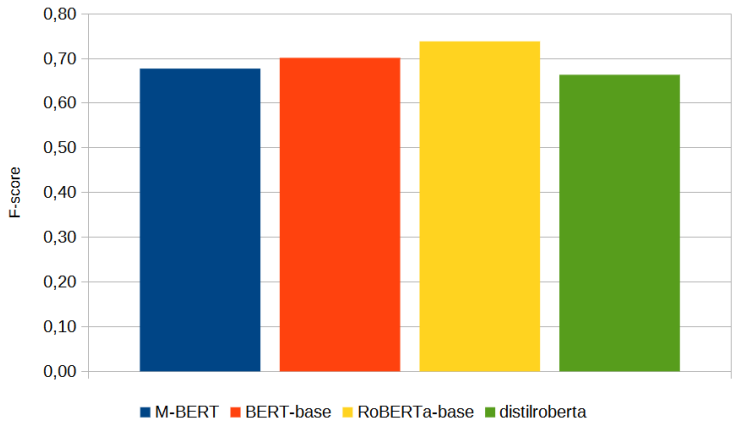
We have introduced the $\text{BERT}_{\text{BASE}}$ and M-BERT language models. There is a wealth of new pretrained Transformer-based language models being released, and we let RoBERTa (Liu et al. 2019) represent the newer models.

RoBERTa shares much of the architecture with BERT and is bidirectional like BERT, which we consider a beneficial for our task. Other model families like GPT are one-directional only. This has proven to be good for text generation, but the model appears to be less versatile, and was not chosen for our task.

RoBERTa vs BERT RoBERTa, *a Robustly Optimized BERT Pretraining Approach* is based on the same architecture as BERT, but a few changes have been made to improve upon the original BERT model:

> The original BERT uses WordPiece tokenizing and a vocabulary of size 30K, while RoBERTa uses a byte-level Byte-Pair Encoding similar to that in GPT-2, and a vocabulary of size 50K.
>
> RoBERTa is trained on a larger dataset, 160GB of text.
>
> RoBERTa is trained with larger batches and for more epochs.
>
> RoBERTa is trained on Masked language modeling over longer sequences, without the training objective of next sentence prediction.

.



Model	F-score
M-BERT	0,6773
BERT-base	0,7013
RoBERTa-base	**0,7382**
DistilRoBERTa	0,6635

Figure 5.2: A comparison between four BERT-based pretrained language models, fine-tuned for targeted Sentiment Analysis in English with the SEMEVAL dataset.

DistilRoBERTa is a faster, simplified version of RoBERTa, inspired by DistilBERT[2]. DistilRoBERTa has 6 layers, while RoBERTa has 12. DistilRoBERTa is twice as fast as RoBERTa, and still has 95% of RoBERTa's performance on GLUE, a benchmark collection for testing automated natural language understanding. Figure 5.2 shows a comparison of performance between four alternative pretrained models, fine-tuned on the task of detecting sentiment target and polarity on the SEMEVAL dataset, tested on its dev set. For our task and with the given parameters,

[2]https://github.com/huggingface/transformers/tree/master/examples/distillation

DistilRoBERTa performed the weakest and RoBERTa$_{BASE}$ performed the best.

5.2.2 Evaluations on NoReC$_{fine}$ MT

We report the experiments on the machine-translated NoReC$_{fine}$ with the RoBERTa pretrained model only, as this model performed best on all English-only experiments. The experiments were also performed on BERT$_{BASE}$ and M-BERT, and RoBERTa was consistently slightly better. The hyperparameters for fine-tuning the models are the same as reported earlier. When fine-tuning a RoBERTa$_{BASE}$ model for Targeted Sentiment Analysis on the machine-translated NoReC$_{fine}$ and evaluating on the test set, we got the results as shown in table 5.1.

Targets and polarity			
Training data	Model	F-score	SD
Norec-fine MT	RoBERTa	0.3166	1.08%

Target boundaries only			
Training data	Model	F-score	SD
Norec-fine MT	RoBERTa	0.3578	1.29%

Table 5.1: Strict evaluation on the machine-translated NoReC$_{fine}$ test set, when fine-tuning on the machine-translated training set.F-score is the average of evaluating 7 models, and SD is their standard deviation.

The experiments on the machine-translated NoReC$_{fine}$ yielded an F_1-score better than our best monolingual Norwegian LSTM-based model, and weaker weaker than the Norwegian model based on M-BERT. The results are carried over to table 5.3 on page 49 for comparison between all approaches.

5.3 Mixed English and Norwegian data

We presented the SEMEVAL dataset in section 3.2. We shuffled the 8,634 Norwegian sentences and the 2,741 English sentences together. A separate version of the dataset was created, with polarity removed from the tags. Since the training data are both Norwegian and English, the pretrained models need to contain both Norwegian and English. This requirement is satisfied in the bilingual fastText word embeddings for the LSTM-based experiments, and in M-BERT for the BERT-based experiments. We fine-tuned models on this mixed training set without changing any hyperparameters, neither for the M-BERT setup or the LSTM setup. We report the F_1-scores from these results in table 5.2 on the next page. The results with and without the added English data are close for both tasks and both architectures. The Norwegian only training set performs best in three out of four setups.

Mixing Norwegian and English training data for Targeted Sentiment Analysis gave no significant improvement or loss in performance . In table 5.3 on page 49, the results from repeated fine-tuning and evaluations on the test set are compared with the other methods presented so far.

Task	Model	Norec-fine	$\text{NoReC}_{\text{fine}}$ + SEMEVAL
Target with polarity	LSTM	0.2710	0.2471
Target with polarity	M-BERT	**0.4077**	0.3950
Target boundaries only	LSTM	0.3641	0.3740
Target boundaries only	M-BERT	**0.5269**	0.5045

Table 5.2: Results from one evaluation on the $\text{NoReC}_{\text{fine}}$ dev set. We see little difference between having only the Norwegian data in the training set, and using mixed English + Norwegian data.

5.4 All experiments compared

Chapter 4 contains our experiments with the Norwegian $\text{NoReC}_{\text{fine}}$ training data only. In this chapter we presented alternative systems for training data, both to mix English and Norwegian training data for a multilingual model, and to machine translate the Norwegian data into English. We here compare the results from both chapters. Table 5.3 on page 49 displays the evaluation on the $\text{NoReC}_{\text{fine}}$ test set. Where training data is "Norec-fine" only, these are the monolingual results from previous chapter carried over.

No gain, no loss We did not gain performance from fine-tuning M-BERT with a mix of $\text{NoReC}_{\text{fine}}$ and SEMEVAL together. But neither did we lose performance. Neither did the machine-translated English version of $\text{NoReC}_{\text{fine}}$ do better when evaluated on the machine-translated $\text{NoReC}_{\text{fine}}$ test set. But performance with this method was better than with our best LSTM-based Norwegian model.

Figure 5.3 on page 49 shows a comparison of the results for the task of extracting sentiment target and polarity. As shown in table 5.3, the tendencies are the same for the task of extracting sentiment target boundaries only. The best performing model for Targeted Sentiment Analysis on $\text{NoReC}_{\text{fine}}$ came from fine-tuning M-BERT_{o}n the $\text{NoReC}_{\text{fine}}$ training set only. The answer for RQ2 is no, adding English training data, or machine translating into English did not improve our system for Targeted Sentiment Analysis on Norwegian text. For $\text{NoReC}_{\text{fine}}$, there are more than eight thousand training sentences in the same language and with the same domain diversity as the testing sets. Adding less than three thousand sentences from a different language and different domain did not help. However, this method may be useful in another resource scenario.

Since Targeted Sentiment Analysis depends on nuances in the language, the task is not well suited for machine translation. Barnes and Klinger (2019) look at how well fine-grained sentiment information is preserved when machine-translating from Basque to English and find that for this language pair, the translated text became a poor resource for Targeted Sentiment Analysis. Saadany and Orasan (2020) show that for user-generated content in Arabic, sentiment is not preserved well in machine translation. For Norwegian, being closer to English in typological features, machine translation into English could be an option to consider. If there were no annotated Norwegian data, Norwegian texts could be machine translated to English and annotated there. A model fine-tuned on these data would be able to perform inference on machine-translated Norwegian text. According to our experiments, this approach could be valuable when annotated target language data are not available.

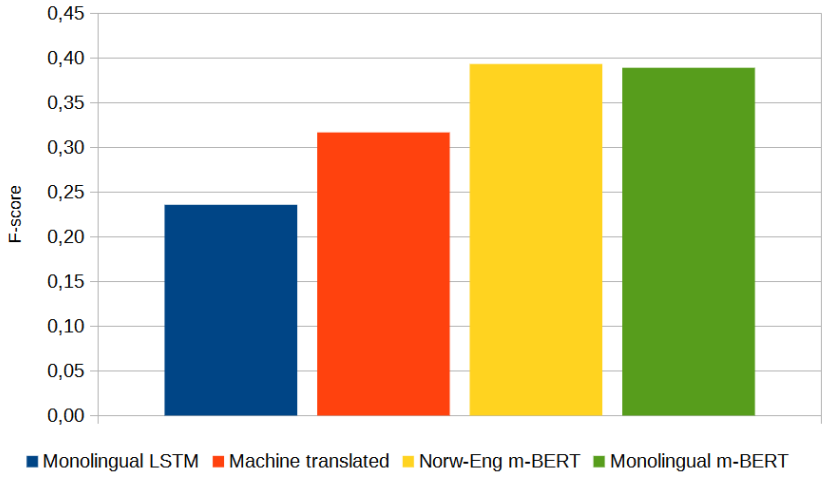


Figure 5.3: **NoReC$_{\text{fine}}$ with English help** Target boundaries and polarity: The new models from this chapter in the middle, compared with the two models from previous chapter on the edges.

Targets and polarity				
	Training data	Model	F-score	SD
Monolingual	Norec-fine	M-BERT	0.3889	1.07%
Monolingual	Norec-fine	LSTM	0.2357	1.80%
New	Norec-fine MT	RoBERTa	0.3166	1.08%
New	Norec-fine + SEMEVAL	M-BERT	**0.3931**	1.00%

Target boundaries only				
	Training data	Model	F-score	SD
Monolingual	Norec-fine	M-BERT	**0.5105**	0.52%
Monolingual	Norec-fine	LSTM	0.3383	1.95%
New	Norec-fine MT	RoBERTa	0.3578	1.29%
New	Norec-fine + SEMEVAL	M-BERT	0.5034	0.62%

Table 5.3: Monolingual models from chapter 4 compared with the new models from this chapter. F-score is the average of evaluating 7 models on the test set. SD is their standard deviation. Mixing Norwegian and English training data for fine-tuning M-BERT gave results similar to using Norwegian data only.

Chapter 6

Experiments with reduced datasets

We have explored different approaches to Targeted Sentiment Analysis for Norwegian text, based on the data available in $\text{NoReC}_{\text{fine}}$. Since these fine-grained sentiment annotations must be done manually by skilled annotators, the dataset is quite expensive to make. In this chapter we look at what can be done with considerably less data. If a language does not have this amount of annotated data, what are the options? This chapter presents evaluations from the same LSTM and M-BERT setups as were used with the full $\text{NoReC}_{\text{fine}}$ dataset. Now, the Norwegian training data are reduced to 25-2000 sentences. This will contribute towards an answer to research question 3a): *Which of the above mentioned approaches might help if the available training data are in the hundreds, and not in the thousands?* Our answer is a Transformer-model that requires only 400 training sentences to surpass a previous state-of-the-art LSTM-based model trained on the full 8634 sentences of $\text{NoReC}_{\text{fine}}$.

No hyperparameters were adjusted for these experiments. Tasks are as before to create individual models for detecting sentiment target and polarity, and for target boundaries only. We were mostly interested in the situation when training sentences are in the hundreds. This would be similar to a situation where one annotated a small amount of data, ran an experiment, annotated more data, reran the experiment etc. The second highest sentence count was 2000, and we did not explore whether performance plateaued between 2000 and the full count of 8634. Before presenting the results with LSTM-based and M-BERT based experiments, we present the scaled-down versions of $\text{NoReC}_{\text{fine}}$, with and without added English data.

6.1 Scaled-down $\text{NoReC}_{\text{fine}}$

We took random samples from the $\text{NoReC}_{\text{fine}}$ training set, to emulate a situation where training data is more sparse. We constructed scaled-down training sets with 25 - 2000 training sentences. All data in a set with lower sentence count are also present in the sets with higher sentence counts.

For each of the reduced datasets, we created a version where the data were mixed with the full SEMEVAL training data. This approach of mixing training data for fine-tuning is inspired by Wu and Dredze (2019) and by Rietzler et al. (2020) who improved their model for Aspect-Based Sentiment Analysis with cross-domain training data. In our experiments we attempted a cross-lingual, cross-domain approach.

6.2 Limited data, LSTM-based models

The pretrained Norwegian-English fastText word embeddings model presented in section 2.2.1 is used for these experiments as well. Figure 6.1 on the next page shows how performance drops when the amount of training data is reduced. The graph shows F_1-scores for the task of detecting target and polarity. Throughout our experiments, the results for detecting target boundaries only show the same tendencies as the results for target boundaries and polarity. All results are present in table 6.1 on page 55.

We see how reduced Norwegian training data leads to reduced performance of the models, evaluated on the dev set. We see the effect of mixing the reduced Norwegian training data with the full SEMEVAL training set. Mixing with the English data creates a smoother line, and increases performance when Norwegian training sentences are 400 and below. The numbers from these experiments are reported in Table 6.1 on page 55.

6.3 Limited data, M-BERT models

We ran the same experiments with the scaled-down datasets, finetuning the pretrained M-BERT model. Evaluation on the $\text{NoReC}_{\text{fine}}$ dev set. No hyperparameters changed from the experiments on the full $\text{NoReC}_{\text{fine}}$ training set. As with the LSTM experiments, we used both the scaled-down Norwegian only training sets and the sets where the same Norwegian sentences are mixed with the entire English SEMEVAL training data.
We see the same tendencies in these experiments as in the LSTM-based experiments. Mixing the reduced Norwegian training data with English helps, especially when the Norwegian data is 400 sentences and below. Although the trend is the same, the values are higher with the BERT-based experiments. The results are illustrated in figure 6.2 for the task of detecting sentiment target and polarity. The F_1-scores are listed in table 6.1.

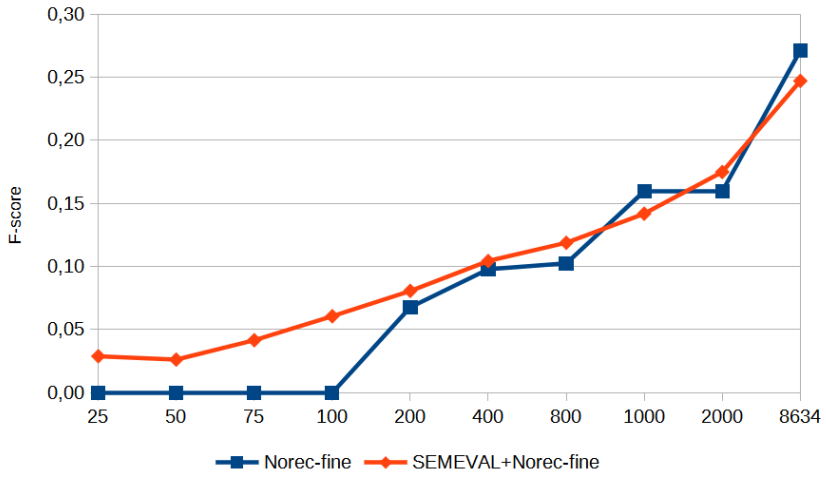


Figure 6.1: **LSTM, boundaries and polarity:** Comparing the effect of reduced training data, and the effect of mixing the full English SEMEVAL training set with the Norwegian data. Task is detecting target and polarity.

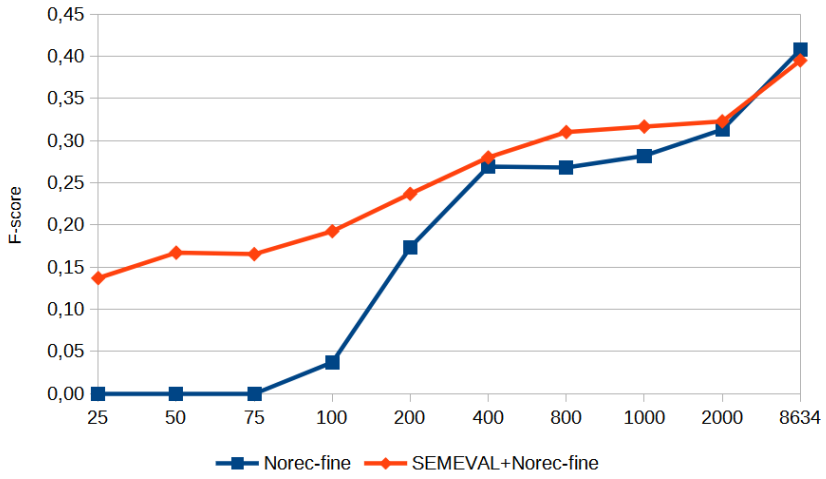


Figure 6.2: **M-BERT, boundaries and polarity:** Fine-tuning M-BERT on subsets of the $\text{NoReC}_{\text{fine}}$ training data, with or without the English SEMEVAL in the mix. From 25 Norwegian sentences up to the full $\text{NoReC}_{\text{fine}}$. Detecting both sentiment target and polarity

6.4 Conclusions from our experiments on scaled-down datasets

We have seen that BERT-based language models perform better than our LSTM-based models for targeted Sentiment Analysis, when fine-tuned on the full $\text{NoReC}_{\text{fine}}$ training set. These experiments show that the M-BERT model is also less sensitive to sparsity in training data than LSTM-based models. See figure 6.3 for a comparison. When fine-tuning on 400 sentences in stead of 8634 sentences, the LSTM-based model's performance dropped to 36% of its performance with 8634 sentences. For M-BERT, the performance at 400 sentences is 66% of its performance at 8634 sentences. This agrees with the findings of Wang et al. (2020a) which shows that with little training data, BERT-based models perform significantly better than LSTM-based models. In our experiments, the BERT-based models always had at least 10 percentage points better F_1-score than the LSTM-based models.

As mentioned, hyperparameter tuning was simpler with the BERT-based models than with LSTM-based models. Although there are many hyperparameters one could have tuned with BERT-based models, starting with default settings worked well, and adjusting these took us from good to better. When training LSTM-based models, it seemed to be much easier to find hyperparameter combinations that worked poorly. Extensive hyperparameter search was needed to go from poor to good performance.

Adding the SEMEVAL English training data gave a noticeable improvement in performance when Norwegian data was scaled down to only a few hundred sentences. When Rietzler et al. (2020) looked at English only data from two domains, they found that finetuning a BERT-based model on data from one domain helped when evaluating on the other domain. We found a similar effect also for cross-domain, cross-lingual datasets. We consider the observations for the range of 100 – 1000 Norwegian sentence to be one of our most important contributions.

To annotate four hundred sentences for sentiment targets and polarity is a much more manageable task than annotating several thousand sentences. We have found that M-BERT fine-tuned for Targeted Sentiment Analysis needs only four hundred sentences to match the performance of our best LSTM-based model trained on all 8634 sentences in $\text{NoReC}_{\text{fine}}$. The added SEMEVAL data contribute to an interesting performance gain when target language training sentences are few. This gain was achieved in spite of the earlier mentioned considerable differences in the datasets SEMEVAL and $\text{NoReC}_{\text{fine}}$. Our answer to research question 3a) about best approaches with limited training data is that we find the multilingual BERT to be the best here as well, and that adding other training data, even cross-domain cross lingual data may help.

Targets with polarity				
Samples	LSTM Norec-fine	LSTM Norec-fine +SEMEVAL	M-BERT Norec-fine	M-BERT Norec-fine +SEMEVAL
25	0	0.0288	0	0.1368
50	0	0.0262	0	0.1670
75	0	0.0415	0	0.1653
100	0	0.0606	0.0368	0.1925
200	0.0676	0.0806	0.1733	0.2369
400	0.0977	0.1043	0.2691	0.2800
800	0.1024	0.1188	0.2681	0.3101
1000	0.1597	0.1418	0.2819	0.3165
2000	0.1598	0.1749	0.3130	0.3228
8634	0.2710	0.2471	0.4077	0.3950

Target boundaries only				
Samples	LSTM Norec-fine	LSTM Norec-fine +SEMEVAL	M-BERT Norec-fine	M-BERT Norec-fine +SEMEVAL
25	0	0.0458	0	0.2043
50	0	0.0628	0	0.2184
75	0	0.0709	0	0.2243
100	0.0068	0.0849	0.1101	0.2379
200	0.0831	0.1200	0.2842	0.3463
400	0.1317	0.1626	0.3540	0.3778
800	0.2120	0.1864	0.4303	0.4291
1000	0.1954	0.2027	0.4385	0.4369
2000	0.2861	0.2392	0.4662	0.4567
8634	0.3641	0.3740	0.5269	0.5045

Table 6.1: F_1-scores for scaled-down versions of $\text{NoReC}_{\text{fine}}$ with and without SEMEVALtraining data. Fine-tuning M-BERT and LSTM models. 8634 samples is the entire training set. Highlighted numbers compare a M-BERT model fine-tuned on 400 Norwegian sentences together with SEMEVAL, and a LSTM-based model trained on the entire $\text{NoReC}_{\text{fine}}$. F_1-score values of 0 means either precision, recall or both are 0.

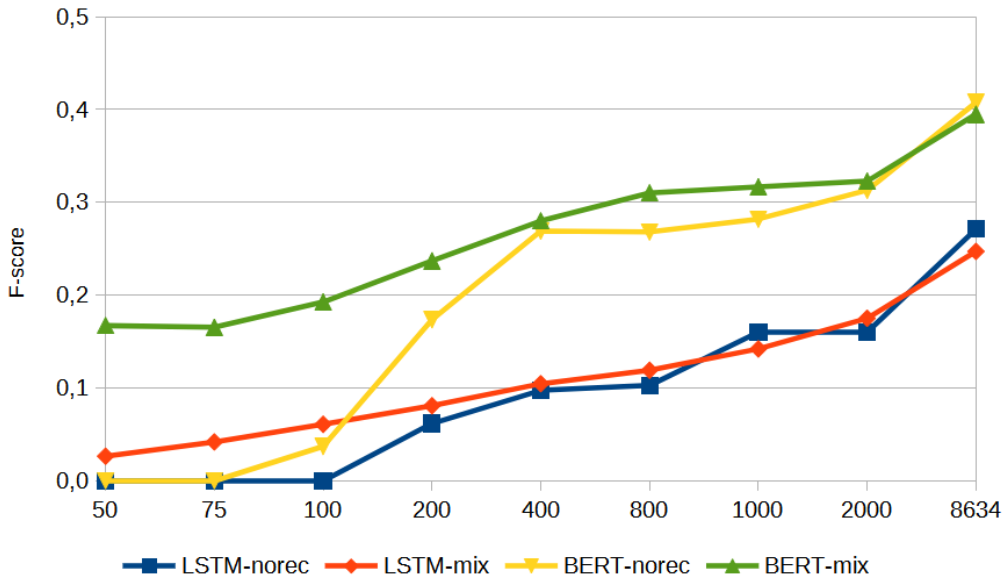


Figure 6.3: Our models for sentiment target and polarity extraction on reduced versions of $\text{NoReC}_{\text{fine}}$. With and without the SEMEVAL dataset mixed into the training data.

Chapter 7

Cross-lingual and cross-domain experiments

Our experiments have shown that adding cross-lingual, cross-domain data may be of help when there is little data available from the same domain and language. In this chapter, we report experiments that compare the importance of language barrier versus domain barrier. We perform zero-shot experiments where no training data are from the same language and domain as the test data. With these experiments, we answer research question 3b): *If there is no training data from the same domain and language, to what degree may data from other domains and languages be useful?* The experiments show how different categories of cross-domain, cross-lingual data contribute to the task of Targeted Sentiment Analysis. We here use the terms *in-domain* for data from the same domain and *in-language* for data from the same language.

7.1 Experimental setup

All experiments in this chapter are based on M-BERT. We use the same hyperparameters as before. All combinations of training set and test set are run three times, and the average F_1-score is reported. The task in all experiments is detecting sentiment target and polarity.

Single-domain training data We joined the train, dev and test segments of $\text{NoReC}_{\text{fine}}$, and split the merged set according to domain. The domains Screen and Music contain more than 2000 sentences, and we sampled 2000 sentences from each for single domain Norwegian training data. We also sampled 2000 sentences from SEMEVAL which is single domain English data. Keeping all three datasets at the same size excluded size differences from influencing the results.

Single domain test data We created Norwegian single-domain test sets from the Restaurants, Screen, and Music domain. The Restaurants domain consists of 340 sentences in the $\text{tNoReC}_{\text{fine}}$ dataset in total. From the Screen

and Music domain, we sampled data not used by the training sets. Due to data scarsity, all three single-domain test sets are in the range of 250 - 350 sentences. This is not much, and we therefore do not compare results in this chapter directly with results from earlier chapters with the dev and test segments of $\text{NoReC}_{\text{fine}}$ consisting of more than 1200 sentences.

7.2 Experiments with data from individual domains

While $\text{NoReC}_{\text{fine}}$ as a whole contains reviews from multiple domains, we have now isolated three single-domain training sets and three single-domain testing sets. Combining these allows for Zero-shot experiments in four categories. The SEMEVAL training set contains restaurants reviews in the English language, and enables cross-lingual experiments, both with in-domain and cross-domain data. The Norwegian in-domain training sets enable in-language experiments, both in-domain and cross-domain. The combinations are listed in table 7.1.

Catecory	**Description**	**Train**	**Test**
CLCD	Cross-lingual cross-domain	SEMEVAL	Music, Screen
CLID	Cross-lingual in-domain	SEMEVAL	Restaurants
ILCD	In-language cross-domain	Music Screen	Screen, Restaurants Music, Restaurants
ILID	In-language in-domain	Music Screen	Music Screen

Table 7.1: Combining the three single-domain training sets and test sets allows for Zero-shot experiments with M-BERT in four categories of distance between train and test set.

7.2.1 Observations from single-domain experiments

We here perform cross-lingual and cross-domain experiments where training and testing data are from one single domain each, in contrast to the multi-domain experiments presented in previous chapters. The results are shown in figure 7.1 and table 7.2

The Cross-lingual experiments more than doubled their performance by moving from cross-domain to in-domain. The cross-lingual cross-domain experiments are fine-tuned on SEMEVAL, containing restaurant reviews only, and evaluated on Norwegian music and screen reviews. When evaluating on the in-domain Norwegian restaurant reviews, F_1-scores increased from 0.09 to 0.18. This indicates that the benefit from auxiliary English cross-domain data observed in chapter 6, would be considerably increased if the data were in-domain.

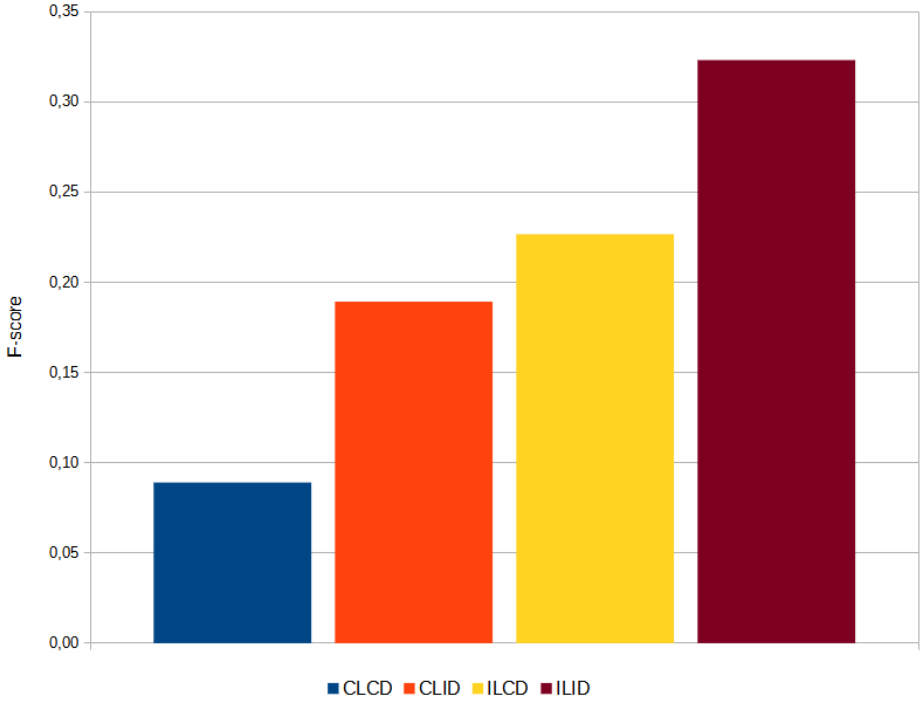


Figure 7.1: Effect of cross-domain and cross-lingual training data: In-domain training yields ten percentage points higher F_1-score, both in-language and cross-lingual. In-language training yields thirteen percentage points higher F_1-score than cross-lingual training.

Category	**Description**	**F-score**
CLCD	Cross-lingual, cross-domain	0.0890
CLID	Cross-lingual, in-domain	0.1893
ILCD	In-language, cross-domain	0.2266
ILID	In-language, in-domain	0.3231

Training data	**Norwegian one domain test sets**		
	Screen	**Restaurants**	**Music**
music	0.2934	0.2331	0.3228
screen	0.3235	0.1723	0.2077
semeval	0.0890	0.1893	0.0890

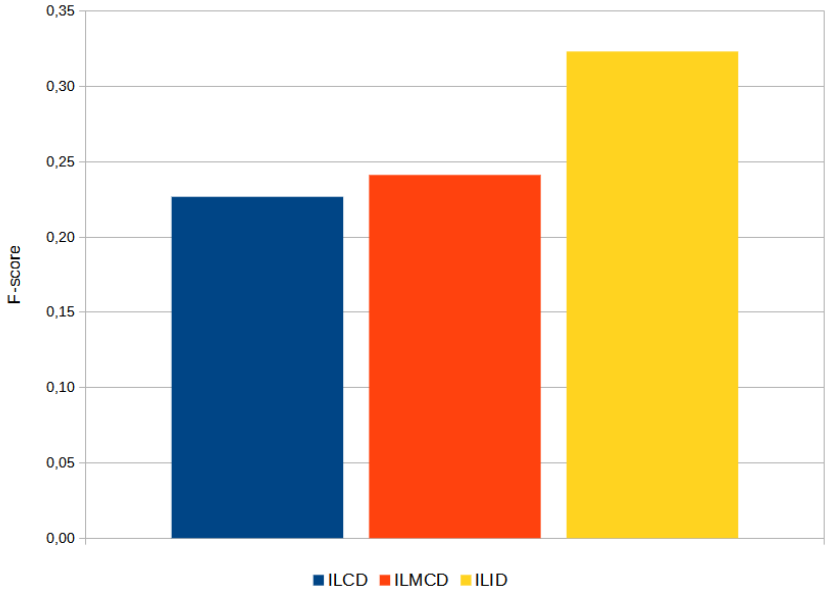
Table 7.2: Training and test data from single domains: From Cross-lingual and cross-domain to in-language, in-domain data. Task is detecting target and polarity. The F_1-scores from the lower part of the table are aggregated in the categories at the upper part of the table.

The Cross-lingual in-domain results were rather close to the in-language cross-domain results. When in-domain, in-language training data are not available, both these combinations are candidates as auxiliary training data. The cross-lingual performance of M-BERT in these experiments corresponds well with the findings of Pires et al. (2019). We conclude that for the Norwegian - English language pair, M-BERT creates a multilingual representation that is a valuable resource for Norwegian NLP in general, and for Targeted Sentiment Analysis in particular.

The superiority of in-language in-domain data comes as no surprise. Moving from in-domain to cross-domain Norwegian data represents a relative performance drop of thirty percent. Whenever obtaining in-language in-domain data is an option, this is considerably better than any of the other options we have explored.

7.3 Mixed-domain training

To investigate further the effect of domain diversity during fine-tuning a model for Targeted Sentiment Analysis, we add a dataset with mixed domains. We sampled 2000 training sentences that were neither from Restaurants, Screen or Music. With this dataset there are no in-domain data in common with the testing sets, but the data are from more than one domain. Figure 7.2 shows that with more domain variety in the training set, the cross-domain performance increases. All training sets are kept at the same size of 2000 sentences. We see the same tendency as reported by Rietzler et al. (2020), that for in-language, cross-domain experiments, using data from multiple domains performs better than data from only one domain. We answer RQ3b): *If there is no training data from the same domain and language, to what degree may data from other domains and languages be useful?* We found in-language cross-domain training data to be the best alternative to in-language in-domain data. Data from multiple domains other than the target domain were slightly better than data from one domain only.



Category	Description	F-score
ILCD	In-language, cross-domain	0.2266
ILMCD	In-language, mixed cross-domain	0.2412
ILID	In-language, in-domain	0.3231

Figure 7.2: Mixed cross-domain training data compared with single-domain data.

Chapter 8

Conclusion and future work

In this thesis we present new models for Targeted Sentiment Analysis in Norwegian. We explore three resource scenarios: A scenario with a relatively large dataset, a scenario with very limited data in the target language, and a cross-lingual and cross-domain scenario where there is no labeled data in the target language and domain.

Chapter 4 For the resource-rich scenario we have developed a new state-of-the-art model for Targeted Sentiment Analysis in Norwegian, based on $\text{NoReC}_{\text{fine}}$. This answer RQ1: *Which neural architecture is best for creating a model for Targeted Sentiment Analysis in Norwegian, based on the NoReC_{fine} dataset?* Our best model was obtained through fine-tuning M-BERT, a Transformer-based model. For the task of detecting sentiment target and polarity, we achieved an F_1-score of 0.389 with M-BERT, and 0.234 with LSTM.

Chapter 5 We answer RQ2: *Are there any multilingual language resources that can improve our Norwegian model for Targeted Sentiment Analysis?* We added English training data to our Norwegian dataset, and this did not improve our results, neither was performance reduced. We found that machine-translating the Norwegian dataset to English reduced performance, but the method yielded better results than the baseline. As machine translation keeps improving, this approach may be relevant for other Norwegian NLP tasks with a different resource scenario.

Chapter 6 We present our best system for Targeted Sentiment Analysis in a scenario with limited data in the target language. We created a mixed dataset from Norwegian and English labeled data. In RQ3 we asked what can be done when amount of training data is limited. RQ3a) is *Which of the tested methods might help if the available training sentences are in the hundreds, and not in the thousands?* Our answer is M-BERT and a bilingual training set. With only 400 Norwegian sentences in the training set, our system performs better on Norwegian test data than an LSTM-based model trained on more than 8000 Norwegian sentences.

Chapter 7 We present our zero-shot experiments with only cross-lingual and cross-domain data. RQ 3b) is: *If there is no training data from the same domain and language, to what degree may data from other domains and languages be useful?* We found in the zero-shot experiments that domain differences between datasets are important, both within one language and in cross-lingual experiments. Our experiments confirm that cross-domain in-language data are better than cross-lingual in-domain data. For cross-domain data in the same language, fine-tuning on data from several domains is better than data from one domain only.

8.1 Future work

8.1.1 Cross-language mixed domain

We found with the zero-shot experiments that domain-similarity is relevant for cross-lingual data as well as for same-language data. We are not aware of any dataset for Targeted Sentiment Analysis that matches the domain diversity of $\text{NoReC}_{\text{fine}}$. But joining more than one single-domain English or other language datasets might create a cross-lingual dataset that is closer in domain mix, and therefore might improve the performance on $\text{NoReC}_{\text{fine}}$. The Restaurants domain is a small category within $\text{NoReC}_{\text{fine}}$. Screen, music and products are the categories with the most sentences in $\text{NoReC}_{\text{fine}}$, and finding cross-lingual training data in one of those domains could shed further light on the cross-lingual same-domain potential of bilingual fine-tuning of multilingual BERT or similar model. Since $\text{NoReC}_{\text{fine}}$ is a relatively large dataset, the experiments may also be reversed so that $\text{NoReC}_{\text{fine}}$ is used as auxiliary data for another language with limited labeled data.

8.1.2 Further pretraining

Before fine-tuning a pretrained language model with labeled data for a certain task, one can continue pretraining the language model itself with unlabeled text in the target language or domain. A second pretraining on task-specific data has also proven helpful (Gururangan et al. 2020). For the task of creating a better model for Targeted Sentiment Analysis on the data in $\text{NoReC}_{\text{fine}}$, the first step of pretraining could be to pretrain on more Norwegian text. Arkhipov et al. (2019) perform such pretraining on four Slavic languages, and obtain new state-of-the-art results for Named Entity Recognition. It is worth noticing, though, that their process took nine days with eight P-100 16Gb GPUs.

A more task-specific pretraining could be pretraining on the entire Norwegian Review Corpus NoReC (Jørgensen et al. 2020). This is the corpus $\text{NoReC}_{\text{fine}}$ was sampled from. Gururangan et al. (2020) find that manually labeled datasets are often a subset of a larger text collection, and that pretraining with this larger text collection consistently improves performance.

8.1.3 Add a new language

The application of M-BERT has been shown to be beneficial for the more than 100 languages that this model was pretrained on. But it might also increase the gap between the Haves and the Have-Nots. Wang et al. (2020b) show a way to extend M-BERT to a new language with an approach that took less than 7 hours to train with a single cloud TPU. It is possible that this could be a method for providing new NLP resources to e.g. the Northern Sámi language.

Bibliography

Abdalla, Mohamed and Graeme Hirst (Nov. 2017). "Cross-Lingual Sentiment Analysis Without (Good) Translation." In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, pp. 506–515. URL: https://www.aclweb.org/anthology/I17-1051.

Adams, Oliver et al. (2017). "Cross-Lingual Word Embeddings for Low-Resource Language Modeling." In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 937–947. URL: http://aclweb.org/anthology/E17-1088.

Agić, Željko et al. (2016). "Multilingual Projection for Parsing Truly Low-Resource Languages." In: *Transactions of the Association for Computational Linguistics* 4, pp. 301–312. URL: http://aclweb.org/anthology/Q16-1022.

Ambartsoumian, Artaches and Fred Popowich (Oct. 2018). "Self-Attention: A Better Building Block for Sentiment Analysis Neural Network Classifiers." In: *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Brussels, Belgium: Association for Computational Linguistics, pp. 130–139. DOI: 10.18653/v1/W18-6219. URL: https://www.aclweb.org/anthology/W18-6219.

Arkhipov, Mikhail et al. (Aug. 2019). "Tuning Multilingual Transformers for Language-Specific Named Entity Recognition." In: *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*. Florence, Italy: Association for Computational Linguistics, pp. 89–93. DOI: 10.18653/v1/W19-3712. URL: https://www.aclweb.org/anthology/W19-3712.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre (July 2017). "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)*. Vancouver, Canada: Association for Computational Linguistics, pp. 451–462. DOI: 10.18653/v1/P17-1042. URL: https://www.aclweb.org/anthology/P17-1042.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). "Neural Machine Translation by Jointly Learning to Align and Translate." In: *CoRR* abs/1409.0473.

Bakliwal, Akshat et al. (June 2013). "Sentiment Analysis of Political Tweets: Towards an Accurate Classifier." In: *Proceedings of the Workshop on Language Analysis in Social Media*. Atlanta, Georgia: Association for

Computational Linguistics, pp. 49–58. URL: https://www.aclweb.org/anthology/W13-1106.

Barnes, Jeremy and Roman Klinger (2019). "Embedding Projection for Targeted Cross-Lingual Sentiment: Model Comparisons and a Real-World Study." In: *J. Artif. Intell. Res.* 66, pp. 691–742.

Baziotis, Christos, Nikos Pelekis, and Christos Doulkeridis (Aug. 2017). "DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis." In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, pp. 747–754. DOI: 10.18653/v1/S17-2126. URL: https://www.aclweb.org/anthology/S17-2126.

Bojanowski, Piotr et al. (2017). "Enriching Word Vectors with Subword Information." In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. DOI: 10.1162/tacl_a_00051. URL: https://www.aclweb.org/anthology/Q17-1010.

Bradley, Margaret M and Peter J Lang (1999). *Affective norms for English words (ANEW): Instruction manual and affective ratings*. Tech. rep. Technical report C-1, the center for research in psychophysiology ...

Conneau, Alexis et al. (2018). "Word Translation Without Parallel Data." In: *ArXiv* abs/1710.04087.

Dave, K., S. Lawrence, and D. Pennock (2003). "Mining the peanut gallery: opinion extraction and semantic classification of product reviews." In: *WWW '03*.

Devlin, Jacob et al. (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423.

Dinu, G. and M. Baroni (2015). "Improving zero-shot learning by mitigating the hubness problem." In: *CoRR* abs/1412.6568.

Dyer, Chris, Victor Chahuneau, and Noah A. Smith (June 2013). "A Simple, Fast, and Effective Reparameterization of IBM Model 2." In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 644–648. URL: https://www.aclweb.org/anthology/N13-1073.

Eberhard, David M., Gary F. Simons, and Charles D. (eds) Fennig (2019). *Ethnologue: Languages of the World. Twenty-second edition*. [Online; accessed 22-May-2019]. URL: http://www.ethnologue.com.

Firth, J. (1957). In: *A Synopsis of Linguistic Theory, 1930-1955*.

Grave, Edouard et al. (May 2018). "Learning Word Vectors for 157 Languages." In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://www.aclweb.org/anthology/L18-1550.

Gururangan, Suchin et al. (July 2020). "Don't Stop Pretraining: Adapt Language Models to Domains and Tasks." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8342–8360. DOI: 10.18653/v1/2020.acl-main.740. URL: https://www.aclweb.org/anthology/2020.acl-main.740.

Heaps, H. (1978). "Information retrieval, computational and theoretical aspects." In:

Hermann, Karl Moritz and Phil Blunsom (June 2014). "Multilingual Models for Compositional Distributed Semantics." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 58–68. DOI: 10.3115/v1/P14-1006. URL: https://www.aclweb.org/anthology/P14-1006.

Hochreiter, S. and J. Schmidhuber (1997). "Long Short-Term Memory." In: *Neural Computation* 9, pp. 1735–1780.

Hsieh, Yu-Lun et al. (Nov. 2016). "How Do I Look? Publicity Mining From Distributed Keyword Representation of Socially Infused News Articles." In: *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*. Austin, TX, USA: Association for Computational Linguistics, pp. 74–83. DOI: 10.18653/v1/W16-6211. URL: https://www.aclweb.org/anthology/W16-6211.

Hu, Minghao et al. (July 2019). "Open-Domain Targeted Sentiment Analysis via Span-Based Extraction and Classification." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 537–546. DOI: 10.18653/v1/P19-1051. URL: https://www.aclweb.org/anthology/P19-1051.

Jørgensen, Fredrik et al. (May 2020). "NorNE: Annotating Named Entities for Norwegian." English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4547–4556. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.559.

Joulin, Armand et al. (2018). "Loss in Translation: Learning Bilingual Word Mapping with a Retrieval Criterion." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Karthikeyan, K. et al. (2020). "Cross-Lingual Ability of Multilingual BERT: An Empirical Study." In: *ArXiv* abs/1912.07840.

Katiyar, Arzoo and Claire Cardie (Aug. 2016). "Investigating LSTMs for Joint Extraction of Opinion Entities and Relations." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 919–929. DOI: 10.18653/v1/P16-1087. URL: https://www.aclweb.org/anthology/P16-1087.

Lample, Guillaume et al. (June 2016). "Neural Architectures for Named Entity Recognition." In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computa-

tional Linguistics, pp. 260–270. DOI: 10.18653/v1/N16-1030. URL: https://www.aclweb.org/anthology/N16-1030.

Levy, Omer and Yoav Goldberg (June 2014). "Dependency-Based Word Embeddings." In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 302–308. DOI: 10.3115/v1/P14-2050. URL: https://www.aclweb.org/anthology/P14-2050.

Li, Xin et al. (2019). "A Unified Model for Opinion Target Extraction and Target Sentiment Prediction." In: *AAAI*.

Liu, Bing (2017). *Sentiment analysis: mining opinions, sentiments, and emotions*. Cambridge University Press.

Liu, Y. et al. (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach." In: *ArXiv* abs/1907.11692.

Luong, Thang, Hieu Pham, and Christopher D. Manning (Sept. 2015). "Effective Approaches to Attention-based Neural Machine Translation." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1412–1421. DOI: 10.18653/v1/D15-1166. URL: https://www.aclweb.org/anthology/D15-1166.

Ma, Xuezhe and Eduard Hovy (Aug. 2016). "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1064–1074. DOI: 10.18653/v1/P16-1101. URL: https://www.aclweb.org/anthology/P16-1101.

Martins, André F. T. et al. (Nov. 2020). "Project MAIA: Multilingual AI Agent Assistant." In: *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. Lisboa, Portugal: European Association for Machine Translation, pp. 495–496. URL: https://www.aclweb.org/anthology/2020.eamt-1.68.

Mihalcea, Rada, Carmen Banea, and Janyce Wiebe (June 2007). "Learning Multilingual Subjective Language via Cross-Lingual Projections." In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 976–983. URL: https://www.aclweb.org/anthology/P07-1123.

Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever (2013a). "Exploiting Similarities among Languages for Machine Translation." In: *CoRR* abs/1309.4168. arXiv: 1309.4168. URL: http://arxiv.org/abs/1309.4168.

Mikolov, Tomas et al. (2013b). "Distributed Representations of Words and Phrases and their Compositionality." In: *CoRR* abs/1310.4546. arXiv: 1310.4546. URL: http://arxiv.org/abs/1310.4546.

Mitchell, Margaret et al. (Oct. 2013). "Open Domain Targeted Sentiment." In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1643–1654. URL: https://www.aclweb.org/anthology/D13-1171.

Nakov, Preslav et al. (June 2016). "SemEval-2016 Task 4: Sentiment Analysis in Twitter." In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 1–18. DOI: 10.18653/v1/S16-1001. URL: https://www.aclweb.org/anthology/S16-1001.

Øvrelid, Lilja et al. (May 2020). "A Fine-grained Sentiment Dataset for Norwegian." English. In: *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 5025–5033. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.618.

Panchendrarajan, Rrubaa and Aravindh Amaresan (Jan. 2018). "Bidirectional LSTM-CRF for Named Entity Recognition." In: *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*. Hong Kong: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/Y18-1061.

Pang, B., Lillian Lee, and Shivakumar Vaithyanathan (2002). "Thumbs up? Sentiment Classification using Machine Learning Techniques." In: *ArXiv* cs.CL/0205070.

Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio (2013). "On the difficulty of training recurrent neural networks." In: *ICML*.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). "Glove: Global Vectors for Word Representation." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://www.aclweb.org/anthology/D14-1162.

Peters, Matthew et al. (June 2018). "Deep Contextualized Word Representations." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://www.aclweb.org/anthology/N18-1202.

Pires, Telmo, Eva Schlinger, and Dan Garrette (July 2019). "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 4996–5001. DOI: 10.18653/v1/P19-1493. URL: https://www.aclweb.org/anthology/P19-1493.

Pontiki, Maria et al. (Aug. 2014). "SemEval-2014 Task 4: Aspect Based Sentiment Analysis." In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, pp. 27–35. DOI: 10.3115/v1/S14-2004. URL: https://www.aclweb.org/anthology/S14-2004.

Pontiki, Maria et al. (June 2015). "SemEval-2015 Task 12: Aspect Based Sentiment Analysis." In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, pp. 486–495. DOI: 10.18653/v1/S15-2082. URL: https://www.aclweb.org/anthology/S15-2082.

Pontiki, Maria et al. (June 2016). "SemEval-2016 Task 5: Aspect Based Sentiment Analysis." In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 19–30. DOI: 10.18653/v1/S16-1002. URL: https://www.aclweb.org/anthology/S16-1002.

Rietzler, Alexander et al. (May 2020). "Adapt or Get Left Behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification." English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4933–4941. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.607.

Ruder, Sebastian, Ivan Vulic, and Anders Søgaard (2017). "A Survey Of Cross-lingual Word Embedding Models." In:

Saadany, Hadeel and Constantin Orasan (2020). "Is it Great or Terrible? Preserving Sentiment in Neural Machine Translation of Arabic Reviews." In: *ArXiv* abs/2010.13814.

Schmitt, Martin et al. (Oct. 2018). "Joint Aspect and Polarity Classification for Aspect-based Sentiment Analysis with End-to-End Neural Networks." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1109–1114. DOI: 10.18653/v1/D18-1139. URL: https://www.aclweb.org/anthology/D18-1139.

Schuster, Tal et al. (June 2019). "Cross-Lingual Alignment of Contextual Word Embeddings, with Applications to Zero-shot Dependency Parsing." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 1599–1613. DOI: 10.18653/v1/N19-1162. URL: https://www.aclweb.org/anthology/N19-1162.

Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting." In: *J. Mach. Learn. Res.* 15, pp. 1929–1958.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum (July 2019). "Energy and Policy Considerations for Deep Learning in NLP." In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3645–3650. DOI: 10.18653/v1/P19-1355. URL: https://www.aclweb.org/anthology/P19-1355.

Sun, C., Luyao Huang, and Xipeng Qiu (2019a). "Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence." In: *ArXiv* abs/1903.09588.

Sun, Chi, Luyao Huang, and Xipeng Qiu (June 2019b). "Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 380–385. DOI: 10.18653/v1/N19-1035. URL: https://www.aclweb.org/anthology/N19-1035.

Taj, Afroz (1997). *Urdu through Hindi: nastaliq with the help of Devanagari*. Rangmahal Press.

Tjong Kim Sang, Erik F. and Fien De Meulder (2003). "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition." In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 142–147. URL: https://www.aclweb.org/anthology/W03-0419.

Turney, Peter (July 2002). "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews." In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 417–424. DOI: 10.3115/1073083.1073153. URL: https://www.aclweb.org/anthology/P02-1053.

Upadhyay, Shyam et al. (Aug. 2016). "Cross-lingual Models of Word Embeddings: An Empirical Comparison." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1661–1670. DOI: 10.18653/v1/P16-1157. URL: https://www.aclweb.org/anthology/P16-1157.

Vaswani, Ashish et al. (2017). "Attention is All you Need." In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Velldal, Erik et al. (May 2018). "NoReC: The Norwegian Review Corpus." In: *Proceedings of the 11th Language Resources and Evaluation Conference*. Miyazaki, Japan: European Language Resource Association. URL: https://www.aclweb.org/anthology/L18-1661.

Vulić, Ivan and Marie-Francine Moens (July 2015). "Bilingual Word Embeddings from Non-Parallel Document-Aligned Data Applied to Bilingual Lexicon Induction." In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, pp. 719–725. DOI: 10.3115/v1/P15-2118. URL: https://www.aclweb.org/anthology/P15-2118.

Wang, Hao et al. (July 2012). "A System for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle." In: *Proceedings of the ACL 2012 System Demonstrations*. Jeju Island, Korea: Association for Computational Linguistics, pp. 115–120. URL: https://www.aclweb.org/anthology/P12-3020.

Wang, Sinong, Madian Khabsa, and Hao Ma (July 2020a). "To Pretrain or Not to Pretrain: Examining the Benefits of Pretrainng on Resource Rich Tasks." In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 2209–2213. DOI: 10.18653/v1/2020.acl-main.200. URL: https://www.aclweb.org/anthology/2020.acl-main.200.

Wang, Yequan et al. (Nov. 2016). "Attention-based LSTM for Aspect-level Sentiment Classification." In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas:

Association for Computational Linguistics, pp. 606–615. DOI: 10.18653/v1/D16-1058. URL: https://www.aclweb.org/anthology/D16-1058.

Wang, Zihan et al. (2020b). "Extending Multilingual BERT to Low-Resource Languages." In: *ArXiv* abs/2004.13640.

Wiebe, Janyce, Theresa Wilson, and Claire Cardie (Jan. 2005). "Annotating Expressions of Opinions and Emotions in Language." In: *Language Resources and Evaluation* 1. URL: https://www.microsoft.com/en-us/research/publication/annotating-expressions-of-opinions-and-emotions-in-language/.

Wolf, Thomas et al. (2019). "HuggingFace's Transformers: State-of-the-art Natural Language Processing." In: *ArXiv* abs/1910.03771.

Wu, Shijie and Mark Dredze (Nov. 2019). "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 833–844. DOI: 10.18653/v1/D19-1077. URL: https://www.aclweb.org/anthology/D19-1077.

Wu, Y. et al. (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." In: *ArXiv* abs/1609.08144.

Yang, Jie, Shuailong Liang, and Yue Zhang (Aug. 2018). "Design Challenges and Misconceptions in Neural Sequence Labeling." In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 3879–3889. URL: https://www.aclweb.org/anthology/C18-1327.

Zhang, Meishan, Yue Zhang, and Duy-Tin Vo (Sept. 2015). "Neural Networks for Open Domain Targeted Sentiment." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 612–621. DOI: 10.18653/v1/D15-1073. URL: https://www.aclweb.org/anthology/D15-1073.

— (2016). "Gated Neural Networks for Targeted Sentiment Analysis." In: *AAAI*.

Ziser, Yftah and Roi Reichart (Oct. 2018). "Deep Pivot-Based Modeling for Cross-language Cross-domain Transfer with Minimal Guidance." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 238–249. URL: https://www.aclweb.org/anthology/D18-1022.