Master's Essay

Adrian Eriksen

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

Master's Essay

Master's Essay

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

Printed: Reprosentralen, University of Oslo

Contents

Chapter 1

Background

1.1 Introduction

In this essay I will explain the challenges of cross-domain sentiment analysis and how we might use transfer learning to solve it. Sentiment analysis is the process of trying to understand the sentiment behind a statement or document using machine learning. This can, among other things, be used to get information from reviews that can provide useful information. There are many different ways to formulate a sentiment. A movie review might state "The movie is not bad at all.". If we simply look for words like "bad" and classify them as negative, we will get inaccurate results.

First, I will explain some of the technologies that has created the foundation for what we now use in language technology.

- Structure of essay -

Pretraining:
BERT
ELMO
task specific

1.2 Sentiment Analysis

Sentiment analysis (SA) is the computational treatment of opinions, sentiments, and subjectivity of texts. SA is also known as opinion mining and a few other terms and has a variety of different applications. It can be used for labeling reviews of movies or books, opinion mining from sites like Twitter.

SA can be done on different levels of text. Document-level SA is the task of classifying the sentiment of a document. The document in this context will be considered as one piece of information, and the score this document receives is determined by the overall sentiment of the author. By applying SA to different documents regarding the same topic, we get a score based on the total number of positive and negative documents.
Sentence-level SA looks at each sentence as positive, negative, or neutral, sometimes with different intensities. When looking at the sentences in a document, there are different levels of subjectivity that can be observed. Some sentences will only state a fact like "The restaurant serves Italian food", while others contain subjective opinions like "The restaurant closes too early". The subjectivity of sentences can affect the intensity, as a sentence based on an opinion or a certain belief, usually indicates a stronger intensity than stating a fact.
Aspect-based SA is different than the two other approaches. Instead of trying to classify a word, sentence, or document as either positive or negative, aspect-based SA is tasked with identifying different aspects associated with a target [Pontiki et al., 2016]. One of the main contributions of aspect-based SA is that in addition to learning whether a review is positive or negative, you also learn which aspects of the review that made it so. If we're looking at a review for a hotel, we could retrieve information like "The breakfast was good", "They never made our bed". With information like this, we can assign the aspect, food, of the target, hotel, has a positive polarity. Likewise, the aspect, service - room, of the target, hotel, has a negative polarity. Being able to extract features like this instead of just "The hotel got a score of 3", is very valuable to most businesses since many consumers share their experiences with products online.

As of today, there are two main approaches to SA, the lexicon-based approach and the machine learning approach. The lexicon-based approach is mostly used on a document-level or sentence-level and uses a lexicon with words or multiword terms. These are usually tagged with sentiment (positive or negative) and sometimes with different intensities (very positive, slightly negative, etc). Given the word or multiword terms, you can further calculate the value of a sentence and then the entire document. One way to do this is by assigning each word a score with either positive or negative numbers while taking negation into account. For example, "The movie was not excellent" should yield a higher score than "The movie was not good", as a strongly polarized word usually reflects a somewhat mixed opinion [Taboada et al., 2011]. One of the benefits of lexicon-based SA is that you don't have any need for labeled data, as the lexicon is pre-defined, and that you get some robustness when applying it on

different domains if the lexicon is well made [Taboada et al., 2011].
The machine learning approach can create a model from a labeled training dataset and then apply it to the target data through standard machine learning methods [Pang et al., 2002], vectorization [Peters et al., 2018, Mikolov et al., 2013, Pennington et al., 2014] or use a premade model and fine-tuning it on the target data [Devlin et al., 2018]. The aforementioned machine learning approaches use unsupervised, semi-supervised and supervised learning. What they all have in common, is that they don't reference any lexicon with pre-defined sentiments. You train an algorithm on graded reviews, and then have it predict the grade given to unseen reviews. Much like the lexicon-based approach, machine learning can be done on both document-level and sentence-level SA.
One final approach that has been used for all granularities of SA is a hybrid between the machine learning and lexicon-based approach [Zhang et al., 2011], where you first train an algorithm on labeled data, before comparing the results with a lexicon to improve accuracy. This approach can also be used for aspect-based SA [Brun et al., 2016]. One of the original challenges with SA was that sentiment is rarely identifiable by keywords alone [Pang et al., 2002]. When humans are presented with the task of selecting a set of keywords to tell whether a movie review is positive or negative, our intuition often leads us towards words like "horrible", "boring" and "sucks" for negative reviews, and "excellent", "thrilling" and "amazing" for positive reviews. As it turns out, selecting words like these gives us a much lower accuracy than if we train a model on labeled reviews, letting the model figure out which words are important.

1.3 Datasets

There is a lot of different datasets that are commonly used for sentiment analysis, covering a variety of domains. The dataset that I will use is in Norwegian, but most of the data used in NLP is in English. In this section, I will discuss some of the most common English ones, before taking a closer look at the one I will use.

Stanford Sentiment Treebank

The Standford Sentiment Treebank [Socher et al., 2013] is a dataset consisting of 11.855 single sentences from movie reviews and fine-grained sentiment labels for 215 thousand phrases. The sentiments are rated between 1 and 25, which makes the annotations very detailed. The dataset has been used as a benchmark to test new language models, as a way to demonstrate high performance.

IMDb Movie Review Dataset

The IMDb Movie Review Dataset [Maas et al., 2011] has close to 50.000 movie reviews, with 25.000 being labeled as positive and 25.000 as negative. It's a dataset for binary sentiment classification, where there are no more than 30 reviews for any given movie. The dataset contains reviews with a score equal to or below 4/10, or a score equal to or above 7/10 so that there are no neutral ratings.

Amazon Review Data

Amazon Review Data [Ni et al., 2019] is a dataset containing 233.1 million product reviews and metadata from Amazon. It includes reviews consisting of text, ratings and helpfulness votes, product metadata consisting of descriptions, category information, price, brand and image features, and links to "also viewed/also bought graphs. The reviews cover 29 different domains including books, music, electronics, video games, beauty, and toys, albeit all of them are Amazon-products.

Twitter

Twitter has long been one of the most important and influential data-source for opinion mining. Already back in 2010, it had millions of users tweeting daily, sharing their opinion on almost everything [Pak and Paroubek, 2010]. The use of hashtags makes the data more easily separable, the word-limit per tweet makes each document concise, and the amount of data grows every day.

NoReC

The Norwegian Review Corpus (NoReC) is a dataset containing more than 35,000 full-text reviews from Norwegian news sources [Velldal et al., 2018].
NoReC covers a range of different domains, including literature, movies, video games, restaurants, music, and theater, in addition to product reviews across a range of categories. Each review is labeled with a score ranging from 1-6, provided by the author of the review. NoReC was primarily created for training and evaluating models for document-level sentiment analysis, which makes it ideal for testing differences between domains on a document-level. The dataset has a good spread between scores, with 3, 4, and 5 being the most frequent. This makes sense, as it usually takes something particularly bad to give a score of 1 or 2, or something extraordinary to give a score of 6. Figure 1.1 shows a distribution of the ratings. Looking at the distributions of the categories, we can see from figure 1.2 that the categories 'screen' and 'music' is the most represented by far. One of the reasons for this is that the different sources label their categories differently. Upon further inspection, we can see from figure 1.3 and figure 1.4 that these actually cover a variety of different categories from the different sources (I have removed the categories 'utenriks', 'kjendis' and 'nyheter' from figure 1.3 because these represented one, four and one reviews respectively).

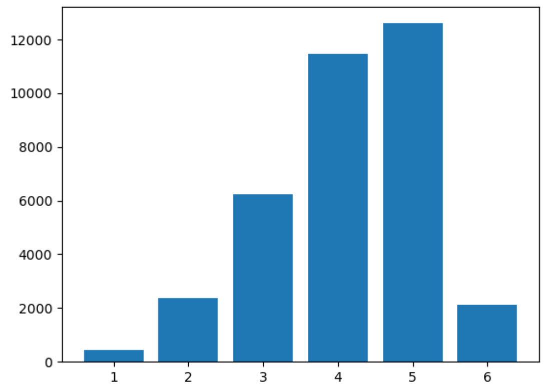

Figure 1.1: Distribution of ratings in the NoReC dataset

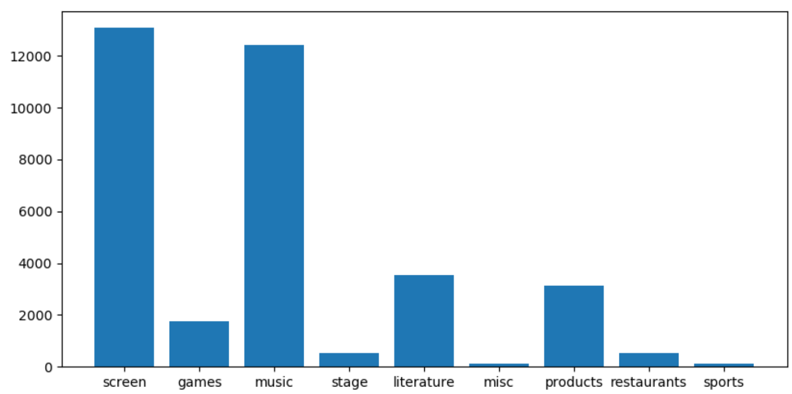

Figure 1.2: The distribution of categories in the NoReC dataset

1.4 Domain Adaptation

Domain adaptation is the task of developing learning algorithms that takes knowledge from labeled data in a source domain and adapt the knowledge to different domains. This is especially interesting in NLP, as we often have a large amount of labeled data in a source domain (e.g IMDb Movie Review Dataset). The prospect of applying the knowledge from an algorithm trained on one domain, to another where we have little to none labeled data, is very appealing. This raises a challenge, however. If we train a model on a specific domain (e.g movie reviews), it transfers poorly to other domains like restaurant reviews.

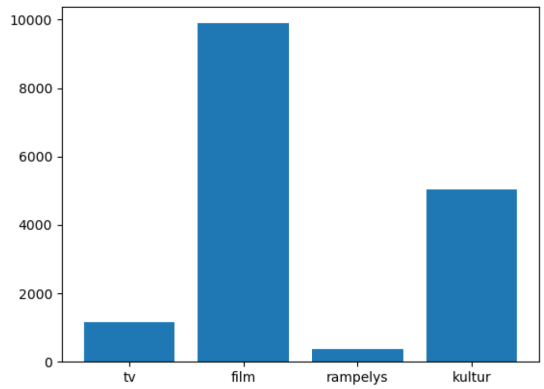

Figure 1.3: The different categories found under 'screen'

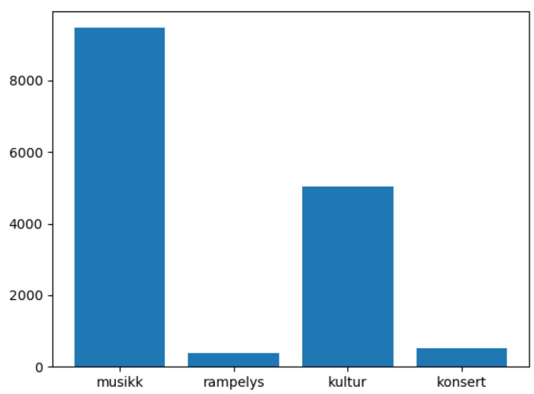

Figure 1.4: The different categories found under 'music'

In the movie review domain, some of the words that carries negative weight is words like "2", "series" and "tv", which makes sense in that specific domain (people tend to disfavor movies based on tv series, and sequels). However, for the restaurant domain, the opposite might be true. If a restaurant has been on tv, or is part of a series, it might be a positive thing. This creates a challenge when we train a model on a specific domain and try to apply it to another.

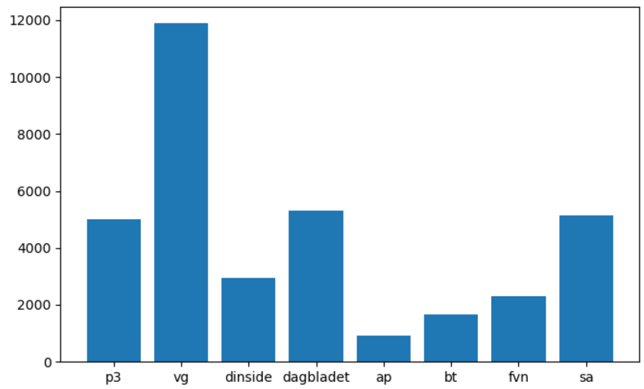

Figure 1.5: The different sources in the NoReC dataset

Transfer Learning

Transfer learning is a means to extract knowledge from a source setting and apply it to a different target setting. If you have a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to improve the performance on D_T using knowledge from D_S and T_S [Pan and Yang, 2010]. According to Pan and Yang, there are three main research issues in transfer learning: 1) what to transfer, 2) how to transfer, and 3) when to transfer. What to transfer, is the task of finding the information that is relevant as well as irrelevant to transfer between the domains. Secondly, we must develop an algorithm that can transfer the information in a satisfactory manner, which is how to transfer. When to transfer is the task of knowing when transfer learning is helpful, and when it's disruptive. Using transfer learning on two completely separate domains may hurt the model's performance[Pan and Yang, 2010].
There are a variety of different types within transfer learning. A taxonomy that shows the variations can be seen in 1.6. In NLP this can be especially useful because words often mean the same in a given context. There are, however, a few different types of transfer learning. One is when you have labeled data in the source domain and adapt the knowledge to different domains, also known as domain adaptation. A different, more common approach, is training on a large amount of unlabeled data, before adapting the representations using self-supervised learning.

Pre-Training

Pre-training in NLP is the task of modeling complex characteristics of word use, and apply this model on several different tasks. Pre-training is usually done

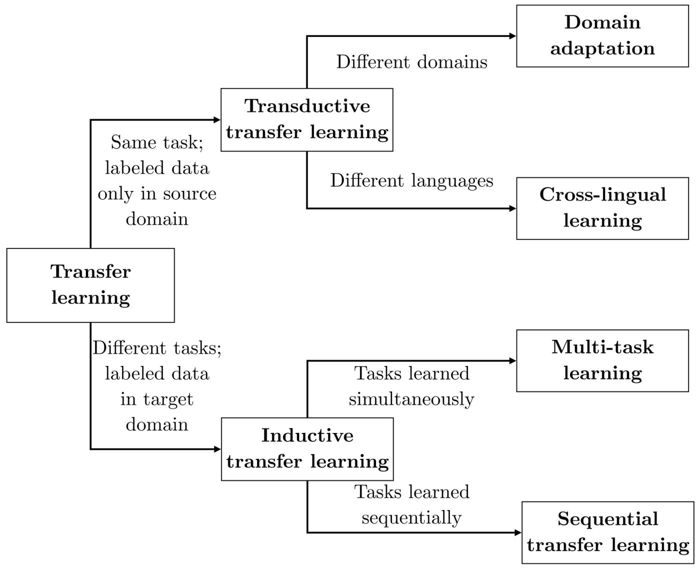

Figure 1.6: A taxonomy for transfer learning in NLP Ruder, 2019

on large amounts of data and has in recent years been used to produce word embeddings. Without going into too much detail, word embeddings aims to capture contexts of words, like "king is to queen as man is to woman". The most commonly used word embeddings can be divided into static and contextualized embeddings. I will first do a short explanation of static embeddings, before discussing contextualized embeddings and finetuning in more detail.

Static Embeddings

Static word embeddings had a breakthrough when Google published their Word2Vec algorithm in 2013 [Mikolov et al., 2013]. This was a breakthrough in NLP, as we now had access to over 1.4 million vectors trained on more than 100 billion words. Word2Vec could be implemented using two different approaches. The first is the continuous bag of words (CBOW), where we try to predict which word is most likely given its context. The second is skip-gram, where we try to predict the context from a word. About a year later, Stanford published their version of static word embeddings called GloVe [Pennington et al., 2014]. In short, what GloVe did differently was that it focused on co-occurrences of words, looking at the probabilities that two words appear together.

Contextualized Embeddings

Contextualized embeddings was the next step in pre-training language models.
In 2018, Embeddings from Language Models (ELMo) was published [Peters et al., 2018].

Where we previously assigned a vector to each word, ELMo looks at the context the word appears in. If we take the word "fall", this could have multiple meanings. One being the verb "to fall", another being the time of year as in "autumn". With traditional embeddings, we would learn the vectors based on a dataset and assign only one vector to "fall". One of the revolutionary things that ELMo did, is that each token is assigned a representation that is a function of the entire input sentence. In other words, the embedding assigned to "fall" is calculated from the sentence it appears in. The way ELMo does this is by using a bidirectional long short-term memory (BiLSTM) RNN to calculate the probability of both previous and future words in the sentence, before returning the contextualized embedding.
Not long after the release of ELMO, Bidirectional Encoder Representations from Transformers (BERT) was published [Devlin et al., 2018]. Upon its release in 2018, it obtained state-of-the-art results on eleven NLP tasks in a variety of fields, using Google's transformer architecture [Vaswani et al., 2017]. Whereas previous language representation models using transformers had been unidirectional [Radford et al., 2018], BERT uses attention mechanisms to learn the contextual relations between all words in a sentence. The way BERT does this is by using a "masked language model" (MLM) pre-training objective. First, the model replaces some of the words in the dataset with the [MASK] token, then the model attempts to predict the actual value of the token, based on the context provided by the unmasked words in the sentence. Next, the model does "next sentence prediction" (NSP). By pairing 50% of the sentences in the dataset, BERT is tasked with predicting whether the next sentence in a document is the next sentence, with a 50% chance it will be. This has proven very useful for tasks like question answering, where models are required to produce fine-grained output at the token level. Upon the release of the paper, Google also released the models used in the paper, BERT_{BASE} and BERT_{LARGE}. These are both incredibly large models with 110M and 340M parameters respectively. Training a model of this size requires an enormous amount of computational power, energy, and time. By making both the code and pre-trained models from the paper publicly available, it became possible for small research groups with limited computational power and funding, to fine-tune BERT and apply it as they saw fit.

Fine-tuning

While both ELMo and BERT can bre trained from scratch, it's more common to use the published models because of the aforementioned time and power it requires. While both models can be used as-is, a common approach is to fine-tune them. Fine-tuning is the task of training a pre-trained model on a new dataset. (Train the entire architecture - Freezing some layers / gradually unfreeze layers - Freeze all layers)

Bibliography

[Brun et al., 2016] Brun, C., Perez, J., and Roux, C. (2016). Xrce at semeval-2016 task 5: Feedbacked ensemble modeling on syntactico-semantic knowledge for aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 277–281.

[Devlin et al., 2018] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[Maas et al., 2011] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

[Ni et al., 2019] Ni, J., Li, J., and McAuley, J. (2019). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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)*, pages 188–197.

[Pak and Paroubek, 2010] Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

[Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

[Pang et al., 2002] Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

[Pontiki et al., 2016] Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S. M., and Eryiğit, G. (2016). SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

[Radford et al., 2018] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

[Socher et al., 2013] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

[Taboada et al., 2011] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

[Velldal et al., 2018] Velldal, E., Øvrelid, L., Bergem, E. A., Stadsnes, C., Touileb, S., and Jørgensen, F. (2018). NoReC: The Norwegian Review Corpus. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference*, pages 4186–4191, Miyazaki, Japan.

[Zhang et al., 2011] Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., and Liu, B. (2011). Combining lexicon-based and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 89.