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 by 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, and determining whether a written text is casual, informative, or friendly, like Grammarly.

As of today, there are two main approaches to SA, the lexicon-based approach and the machine learning approach. The lexicon-based approach 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. Lexicon-based SA is mostly used on a document-level or sentence-level. 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 recieves is determined by the overall sentiment of the author. By applying SA to a number of 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.
A The machine learning approach could either involve creating a model from a labeled training dataset and then applying it to the target data, or use a premade model and fine-tuning it on the target data. Both of the aforementioned machine learning approaches use supervised learning, where you train an algorithm on a large number of graded reviews and then have it predict the grade given to unseen reviews. One could also do a hybrid between the two, where you first train an algorithm on labeled data, before comparing the results with a lexicon to improve accuracy. 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. This, however, raises a challenge when it comes to domains. If we train a model on a specific domain (e.g movie reviews), it transfers poorly to other domains like restaurant reviews. 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), but this is not applicable while trying to predict whether a review of a restaurant is positive or negative.

1.3 NoReC

The Norwegian Review Corpus (NoReC) is a dataset containing more than 35,000 full-text reviews from Norwegian news sources [Velldal et al., 2017]. 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, 1.2 and 1.3 shows a distribution of the data through the training set, development set and test set. As we can see from the plots, the proportions of the scores are relatively equal in all three, which is important when developing a model.

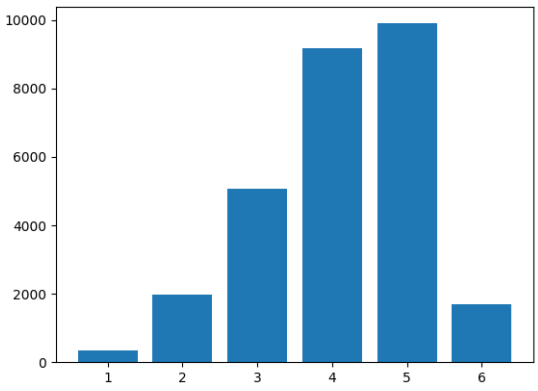

Figure 1.1: training

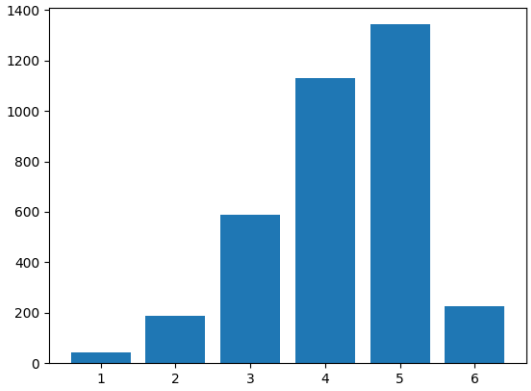

Figure 1.2: development

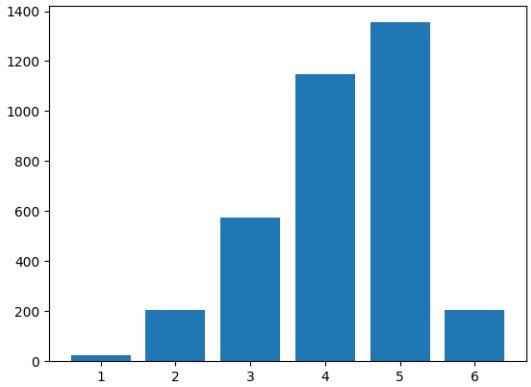

Figure 1.3: test

1.4 Transfer Learning

Transfer learning is a means to extract knowledge from a source setting and apply it to a different target setting. For example, one could train a model to recognize dogs, and then apply the knowledge to a model trying to recognize wolves. 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.

1.5 Pretraining

The large amount of information that each word could contain, turned out to be solvable by vectorization (embedding). It was discovered that by mapping each word to a vector, we could keep a lot of its properties without having to process all of its information. By looking at which words appeared in the same context, we could place synonyms close to each other in the vector space. If we take the example of the words "king" and "queen", they would be placed close to each other in the vector space, along with words like "royalty" and "palace". However, "queen" would be closer to "female", while "king" would be closer to "male". One of the main problems remained, however. Training the embeddings on a large dataset is still very expensive in both time, energy, and resources.

Contextualized Embeddings

ELMo - Embeddings from Language Models was the next step in the evolution of word vectorization [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.

Finetuning

BERT - Bidirectional Encoder Representations from Transformers is probably the most influential invention in NLP in recent years [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. Whereas previous language representation models like OpenAI GPT had been unidirectional, BERT uses attention mechanisms to learn the contextual relations between words. 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.

1.6 Domain Adaptation

Domain Adaptation is the part of transfer learning where you want to apply the model trained during the pretraining to a target domain.

Domain Adaptation for Sentiment Analysis

Bibliography

[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.

[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.

[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*.

[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.

[Velldal et al., 2017] Velldal, E., Øvrelid, L., Bergem, E. A., Stadsnes, C., Touileb, S., and Jørgensen, F. (2017). NoReC: The norwegian review corpus. Common Language Resources and Technology Infrastructure Norway (CLARINO) Bergen Repository.