Sentiment Analysis: Game Review

Md Salman Rahman, Prosanta Barai, Harrinson Arrubla

Data Exploration

DATA: 2,000 reviews which 1000 of those belong to the training data.

MOST REPEATED WORD: “The” with more than 800 repeated times.

FIRST MODEL.

MODEL.

The natural approach is by using the ntkl python library methodology set by Steven Bird, Ewan Klein & Edward Loper in their book Natural Language Processing with Python.

Processing Raw Text

1. **Normalization.** Applying the lower case function to all the data such that words as *The* and *the* will not be different (ntkl).
2. **Tokenization.** Process to break up the strings into words and punctuation (ntkl).
3. **Stopping Words.** filter out of a document high-frequency words such as the, to (nltk.corpus - stopwords).

4. Stemmers. Removing the suffix from a word and reduce it to its root word. Basic stemmers: - Porter (indexing) and Lancaster (fast and sophisticated algorithm).
5. Remove. Removing the non abecedary characters (re).

IMPLEMENTATION.

Using the Sentiment Intensity Analyzer, split the data in between positive and negative to later on classify by zeros and ones (ntkl.sentiment - SentimentIntensityAnalyzer).

Machine Learning for Text

Methods

Machine Learning!

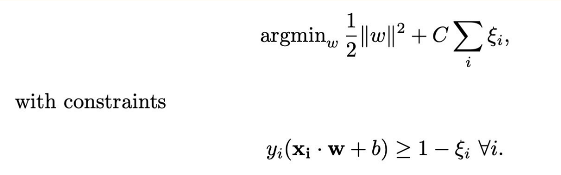
- Support Vector Machine
- Naive Bayes Classifier
- Multinomial Bayes

The Support Vector Machine

1. The Support Vector Machine is a classifier, originally proposed by Vapnik

2. Finds a maximal margin separating hyperplane between two classes of data

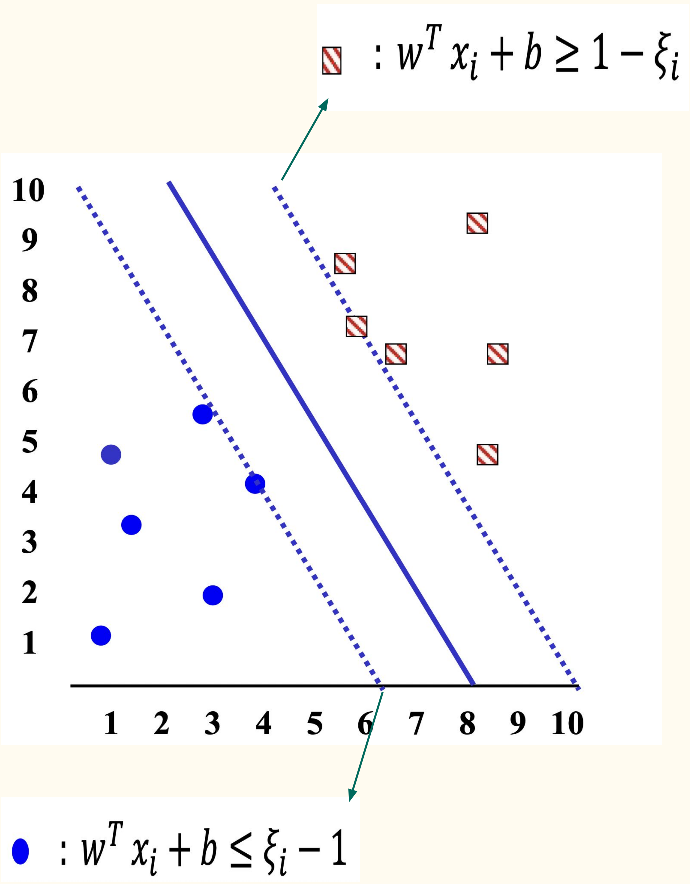
3. Following optimization problem:



$$\text{argmin}_w \frac{1}{2}\|w\|^2 + C\sum_i \xi_i,$$

with constraints

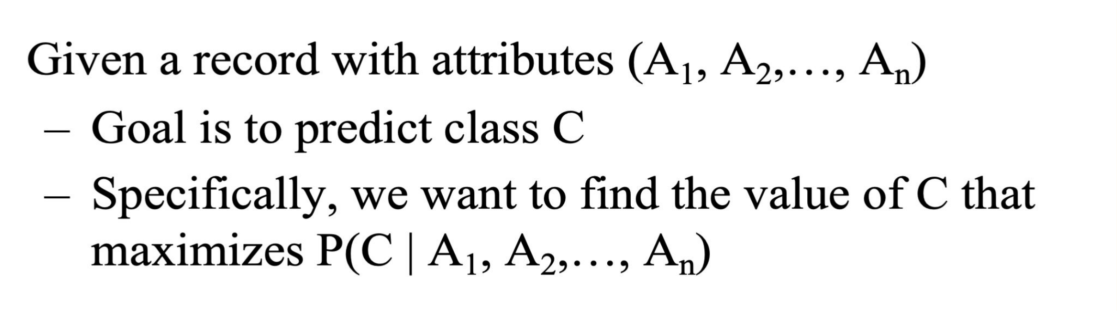
$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) \geq 1 - \xi_i \ \forall i.$$



▨ : $w^T x_i + b \geq 1 - \xi_i$

● : $w^T x_i + b \leq \xi_i - 1$

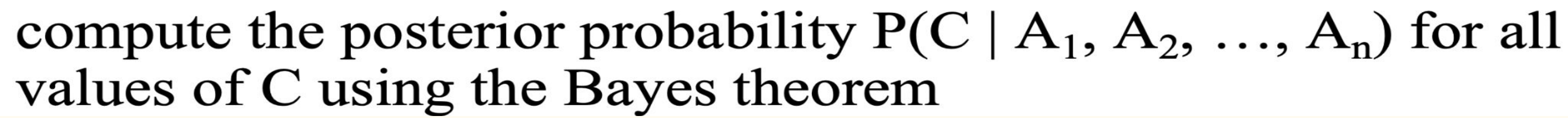
Naive Bayes Classifiers

1. Naive Bayes is a simple Bayesian text classification algorithm.
2. Assumes that each term in a document is drawn independently from a independent Gaussian probability distribution and classifies according to the Bayes optimal decision rule.
3. 

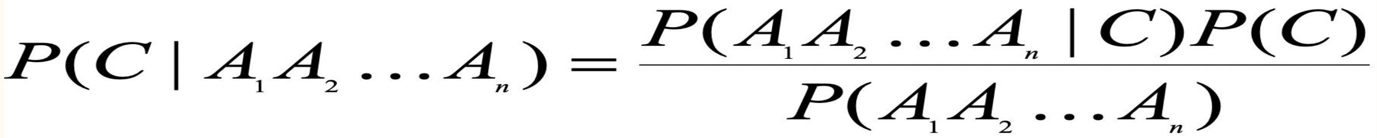
   Given a record with attributes ($A_1, A_2, \ldots, A_n$)

   – Goal is to predict class C

   – Specifically, we want to find the value of C that maximizes $P(C \mid A_1, A_2, \ldots, A_n)$

4. 

   compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of C using the Bayes theorem



$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

Multinomial Naive Bayes

1. The *multinomial Naive Bayes* or *multinomial NB* model, a probabilistic learning method.
2. The probability of a document d being in class c is computed as.

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

Where P(t_k|c) is the conditional probability of term t_k occurring in a class c.

The only difference with the previous GNBC is the way prior probability is calculated.

$$\hat{P}(c) = \frac{N_c}{N},$$

Performance

Performance

1. SVM Can control the model complexity by providing the control on cost function, margin parameters to use. SVM and MNBC performed almost the same.

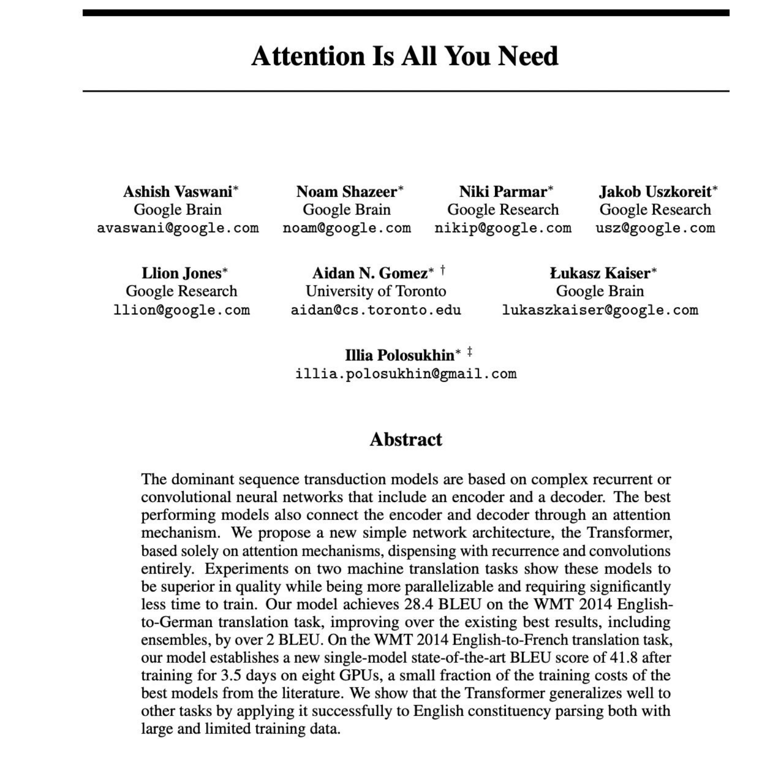
2. The Gaussian NBC performed the worst. It might be due the violation of the normality assumption.

Model	Accuracy
SVM	0.73989
GNBC	0.57575
MNBC	0.74494

Transformer

Transformer

A transformer is a deep learning model that uses the self-attention mechanism to weigh the importance of each element of the input data differently. It is utilized mainly in natural language processing (NLP) and computer vision applications (CV).



Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*] [†]
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

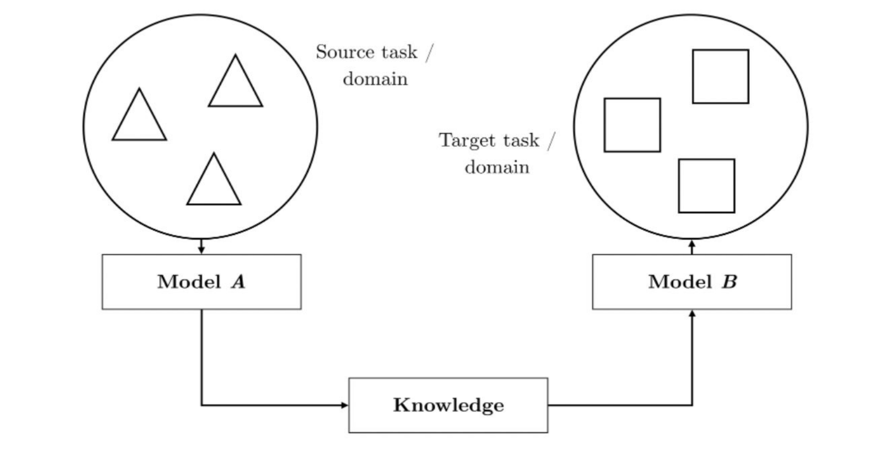
Illia Polosukhin[*] [‡]
illia.polosukhin@gmail.com

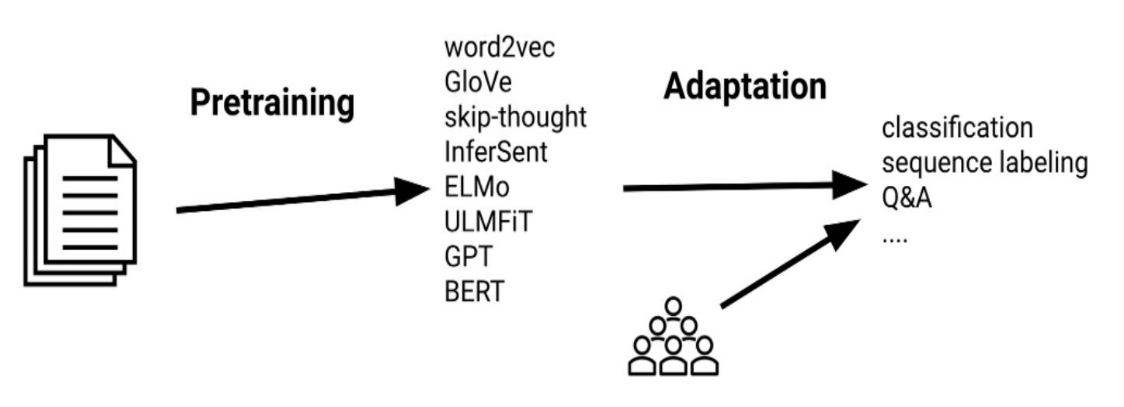
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Problem with Transformer

Author of self attention paper trained the models on one machine with 8 NVIDIA P100 GPUs. For base models using the hyperparameters described throughout the paper, each training step took about 0.4 seconds. They trained the base models for a total of 100,000 steps or 12 hours. For the big models,(described on the bottom line of table 3), step time was 1.0 seconds. The big models were trained for 300,000 steps (3.5 days).



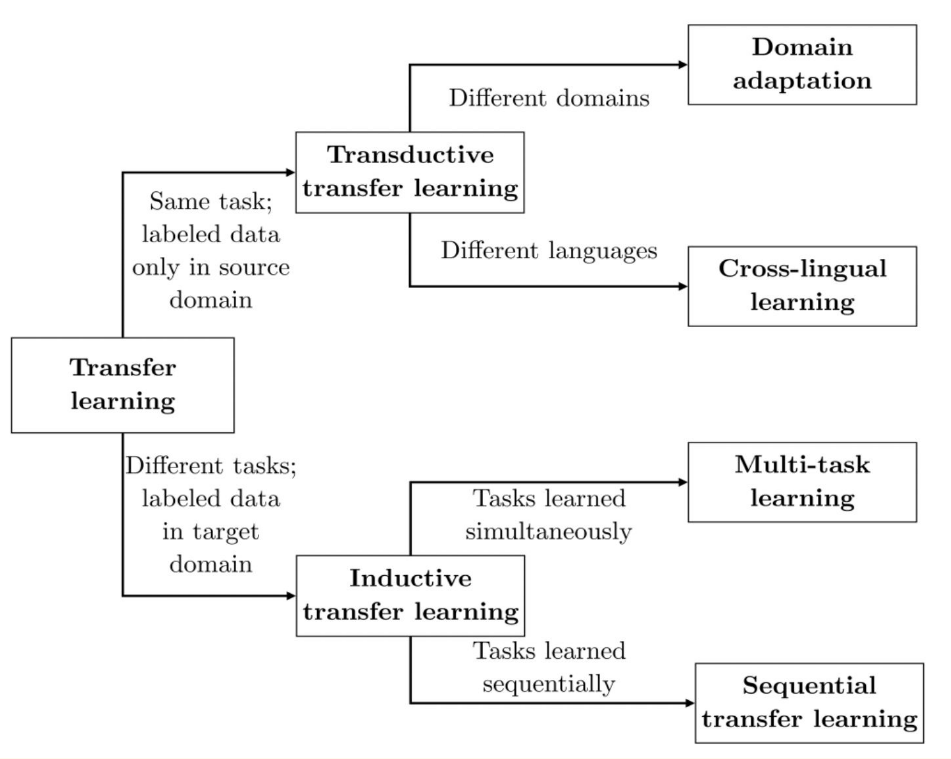


Transfer Learning

Transfer Learning Task

1. Three common transfer learning a) whether the source and target settings deal with the same task; and b) the nature of the source and target domains; and c) the order in which the tasks are learned

2. Common procedure is to pretrain representations on a large unlabeled text corpus using your preferred method, and then adapt these representations to a supervised target task using labelled data.


Transfer learning
Same task; labeled data only in source domain
Transductive transfer learning
Different domains
Domain adaptation
Different languages
Cross-lingual learning
Different tasks; labeled data in target domain
Inductive transfer learning
Tasks learned simultaneously
Multi-task learning
Tasks learned sequentially
Sequential transfer learning

Pre Training Language Model

1. One explanation for language modeling effectiveness could be that it is a difficult endeavor, even for humans. A model must learn about syntax, semantics, and certain facts about the world in order to have any hope of performing this task. A model can do a reasonable job if given adequate data, a large number of parameters, and enough computing power.

2. According to a recent PRD analysis of human language (Hahn and Futrell, 2019), human language—and language modeling—have infinite statistical complexity, although it can be approximated well at lesser levels. This finding has two implications: 1) we can get good outcomes with relatively tiny models, and 2) scaling up our models has a lot of potential.

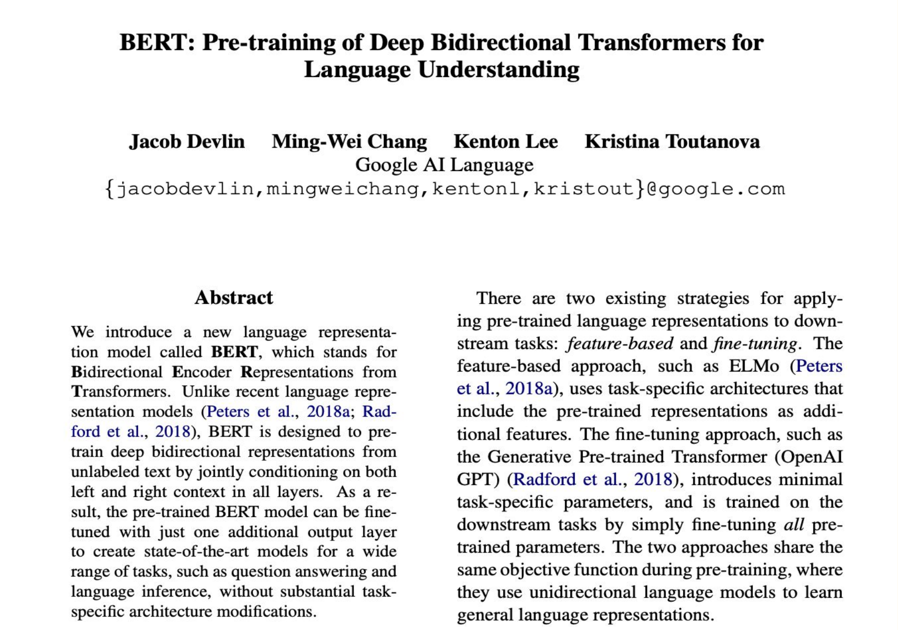
Why does language modelling work so well?

Methods

DistilBERT!(distilled version of BERT)

- smaller
- faster
- cheaper
- lighter

Pretrained and Modifications



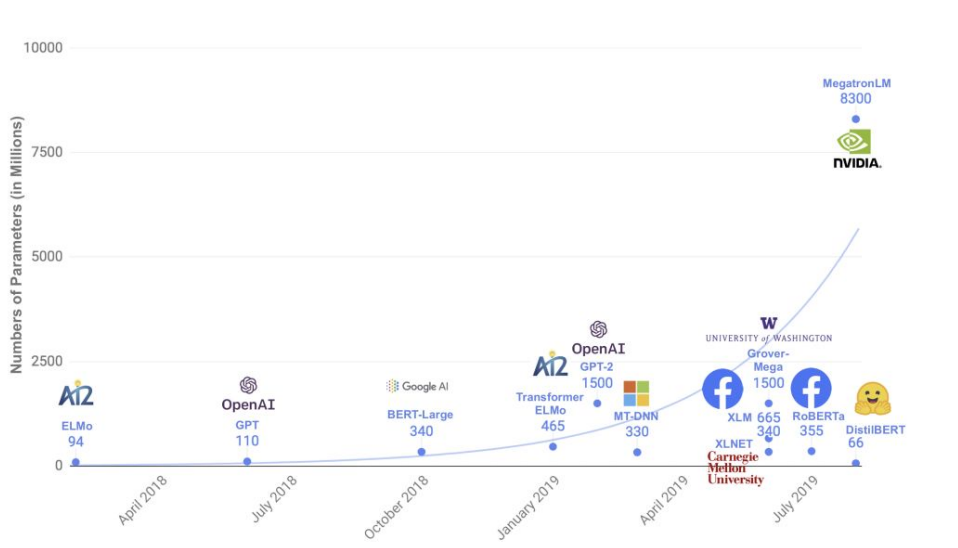
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pretrained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.



Efficiency of DistilBERT

DistilBERT(pre trained with three objectives)

1. Distillation loss
2. Masked language modeling (MLM)
3. Cosine embedding loss

Performance

Model Performance

Pretrained Model	Accuracy
distilbert-base-uncased-finetuned-sst-2-english	0.91919
distilbert-base-uncased	0.90656
siebert/sentiment-roberta-large-english	0.86868

Conclusion

In case of computational limitation pre training and task adaptation technique proven to be very effective.

Using the simple architecture leads to better accuracy (avoiding overfitting) reducing the computational complexity.

Reference

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.