

Student Thesis

Level: Bachelor

**Coronavirus public sentiment analysis with BERT deep learning**

Description: Streck

Author: Jintao Ling

Supervisor: Joonas Pääkkönen & Xiaoyun Zhao Examiner: Pär Eriksson

Subject/main field of study: Informatics Course code: GIK28T

Credits: 15.0 hp

Date of examination: 2020-05-20

|  |  |
| --- | --- |
| At Dalarna University it is possible to publish the student thesis in full text in DiVA. The publishing is open access, which means the work will be freely accessible to read and download on the internet. This will significantly increase the dissemination and visibility of the student thesis.  Open access is becoming the standard route for spreading scientific and academic information on the internet. Dalarna University recommends that both researchers as well as students publish their work open access.  I give my/we give our consent for full text publishing (freely accessible on the internet, open access): | |
| Yes ☒ | No ☐ |

Dalarna University – SE-791 88 Falun – Phone +4623-77 80 00

**Abstract**

Microblog has become a central platform where people express their thoughts and opinions toward public events in China. With the sudden outbreak of coronavirus, the posts related to coronavirus are usually followed by a burst immediately in microblog volume, which provides a great opportunity to explore public sentiment about the events. In this context, sentiment analysis is helpful to explore how coronavirus affects public opinions.

Deep learning has become a very popular technique for sentiment analysis. This thesis uses Bidirectional Encoder Representations from Transformers (BERT), a pre-trained unsupervised language representation model based on deep learning, to generate initial token embeddings that are further tuned by a neural network model on a supervised corpus, a sentiment classifier is constructed. We utilize data recently made available by the government of Beijing which contains 1 million blog posts from January 1 to February 20, 2020. Also, the model developed in this thesis can be used to track the sentiment variation with Weibo microblog data in the future.

At the final stage, the variation of public sentiment is analyzed and presented with visualization charts of preformed people sentiment variation with the development of coronavirus in China. Comparison of the results between labeled data and all data is performed in order to explore how thoughts and opinions evolve in time. The result shows a significant growth of the negative sentiment on January 20 when a lockdown started in Wuhan, and afterward the growth becomes slower. Around February 7 when doctor Wenliang Li died, the number of negative sentiments reached its peak.

**Keywords:** coronavirus, deep learning, sentiment analysis, token embedding, social media.

# Table of Content

1. [Introduction 1](#_bookmark0)
   1. [Background 1](#_bookmark1)
   2. [Research Question 2](#_bookmark2)
   3. [Contribution 2](#_bookmark3)
   4. [Structure of Thesis 2](#_bookmark4)
2. [Theoretical background 3](#_bookmark5)
   1. [Deep Learning 3](#_bookmark6)
   2. [Pre-trained Language Models 4](#_bookmark8)
   3. [Sentiment Analysis (SA) 5](#_bookmark9)
   4. [Social Media Analytics 8](#_bookmark10)
3. [Research Methodology 10](#_bookmark12)
   1. [Literature Review 10](#_bookmark13)
   2. [Research Strategy 10](#_bookmark14)
   3. [Data Generation 10](#_bookmark15)
   4. [Data Analysis 10](#_bookmark16)
   5. [Experiment Process 10](#_bookmark17)
4. [Data 12](#_bookmark20)
   1. [Data Exploration 12](#_bookmark21)
   2. [Data Preprocessing 13](#_bookmark24)
5. [Method 17](#_bookmark28)
   1. [Model Architecture 17](#_bookmark30)
   2. [Implementation 19](#_bookmark33)
   3. [Performance Metrics 20](#_bookmark35)
6. [Results 24](#_bookmark42)
   1. [Sentiment KDEs 24](#_bookmark43)
   2. [Waterfall Chart 25](#_bookmark45)
   3. [Sentiment Waterfall Charts 28](#_bookmark49)
   4. [Discussion 28](#_bookmark50)
7. [Conclusions 30](#_bookmark51)
8. [References 31](#_bookmark52)

# Introduction

## Background

The outbreak of the Corona Virus Disease 2019 (COVID-19) has spread rapidly across the world. By May 1, 2020, there are 3,682,968 confirmed cases all over the world. This pandemic has a caused aroused wide public concern in China. As the coronavirus broke out in China so suddenly, knowing the public sentiment variation is helpful for the government to handle and control the pandemic development, as well as to support more scientific and effective work. For example, is it feasible and acceptable for people about lockdown Wuhan as it is the first time happened in China. Sentiment analysis is a useful technique to quick acquire people’s insights using large volumes of text data. From sentiment analysis, clear feedback about government policies on coronavirus from the public can be present, which is really important information for government decision making for pandemic control. However, it is impossible to ask everyone how they feel about coronavirus as there are 1.4 billion people in China currently. At this time, the sentiment classification technique makes it possible to explore public sentiment variation.

Because of the increasing power of social networks for expressing opinions about hot events and the rapid spreading of online content, online opinions have become a very valuable asset for expressing opinions and providing data for sentiment analysis. Online opinions can be regarded as sentiment, which means a view, an opinion or an emotion that is expressed. In this context, the data on social media platforms have become a valuable asset for sentiment analysis research. Nowadays, more and more research and concentration are put on Sentiment Analysis (SA), whose goal is the classification of opinions and sentiments expressed in the text generated by a human party. The data that exists on social networks, especially those that describe people's views, ideas and comments, has recently been amplified.

Understanding and predicting people’s sentiments may affect various decisions made by a government. For instance, if we know that people have reacted to an event, such as the lockdown of Wuhan, very negatively, then we might want to reconsider locking down other cities. Also, it is helpful understand how people feel about events that are outside of the control of the government because the government could hinder the spreading of this piece of news, for example, if the government wishes to hinder the spread of information that might upset people. Conversely, the government may want to spread information that is expected to make people feel positive, for example, to encourage or uplift people.

## Research Question

Since the rapid development of sentiment analysis techniques and the massive posts followed by the coronavirus outbreak, exploration about public sentiment variation becomes possible. Otherwise, combining with the big news and events that happened related to coronavirus, the possible reason which may cause the public sentiment varied can be found. In this thesis, the following two research questions are answered:

* How public sentiments toward coronavirus evolved from January 1 to February 20, 2020?
* What events affect public sentiment?
* How well does BERT deep learning perform in coronavirus sentiment classification?

## Contribution

The primary contribution of this study is studying the effects of coronavirus on public sentiment over time. Otherwise, a complete and detailed sentiment analysis project using a pre-trained language model is presented. For future researchers who want to track the public sentiment variation about coronavirus, the model developed in this thesis can be reused.

## Structure of Thesis

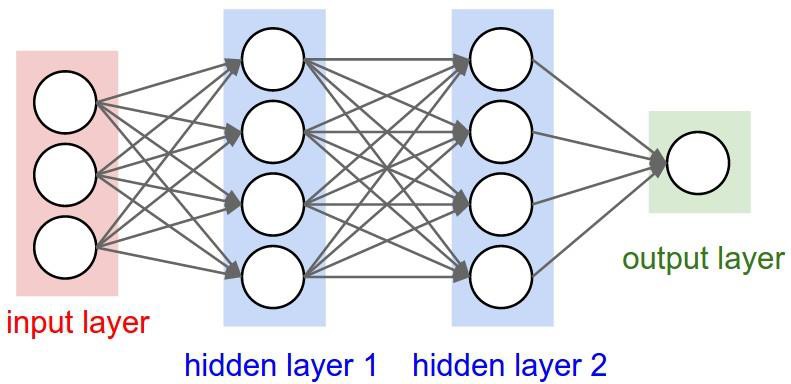
The thesis begins with a literature review in [Section 2,](#_bookmark5) which explores the paradigms of deep learning, social media analytics, and sentiment analysis. The data used in this study are described in [Section 3](#_bookmark19). [Section 4](#_bookmark29) describes the methodology used, and [Section 5](#_bookmark42) displays the results in the form of a line chart and clustered bar chart and discuss the drawbacks of this thesis with some optimization suggestions. The summary part concludes the work of this thesis in [Section 6.](#_bookmark51)

# Theoretical background

## Deep Learning

In the 1980s, Noel Entwistle and colleagues proposed the term ‘deep learning’ for the first time in research about how to distinguish deep and surface learning (Entwistle and Ramsden, 1983). Artificial neural networks have become a branch of machine learning (Schmidhuber, 2015), deep learning can be divided into supervised, semi-supervised or unsupervised learning (LeCun, Bengio and Hinton, 2015).

1980'lerde Noel Entwistle ve meslektaşları, derin ve yüzeysel öğrenmenin nasıl ayırt edileceğine ilişkin araştırmalarda ilk kez 'derin öğrenme' terimini önerdiler (Entwistle ve Ramsden, 1983). Yapay sinir ağları makine öğreniminin bir dalı haline gelmiştir (Schmidhuber, 2015), derin öğrenme denetimli, yarı denetimli ve denetimsiz öğrenme olarak ikiye ayrılabilir (LeCun, Bengio ve Hinton, 2015).

The main concept in deep learning algorithms is the automated extraction of representations from data (LeCun, Bengio and Hinton, 2015). Another key concept closely related to deep learning method is learning the distributed representation of data. In this case, each sample can be represented compactly, leading to a richer generalization.

**Figure 3**. The architecture of deep learning.

Derin öğrenme algoritmalarındaki ana kavram, verilerden temsillerin otomatik olarak çıkarılmasıdır (LeCun, Bengio ve Hinton, 2015). Derin öğrenme yöntemiyle yakından ilgili diğer bir anahtar kavram, verilerin dağıtılmış temsilini öğrenmektir. Bu durumda, her örnek daha zengin bir genellemeye yol açacak şekilde kompakt bir şekilde temsil edilebilir.

For deep learning algorithms, it can be simply thought of as deep architectures of consecutive layers (see [Figure 3](#_bookmark7)). Each layer applies nonlinear transformation to its input and output the representation. The input layer contains your input data. The hidden layer is trying to learn different aspects of the data by minimizing an error or cost function. The output layer consists of the output data. The purpose is to learn a complex and abstract data representation hierarchically by passing data through multiple transformation layers (Najafabadi, et al., 2015). The input data are fed to the first layer such as token (word piece) embeddings. The output of each layer is the input of the next layer.

Derin öğrenme algoritmaları için, basitçe ardışık katmanların derin mimarileri olarak düşünülebilir (bkz. Şekil 3). Her katman, girdisine doğrusal olmayan dönüşüm uygular ve gösterimi çıkarır. Giriş katmanı, giriş verilerinizi içerir. Gizli katman, bir hatayı veya maliyet fonksiyonunu en aza indirerek verilerin farklı yönlerini öğrenmeye çalışıyor. Çıktı katmanı çıktı verilerinden oluşur. Amaç, verileri çoklu dönüşüm katmanlarından geçirerek karmaşık ve soyut bir veri gösterimini hiyerarşik olarak öğrenmektir (Najafabadi, vd., 2015). Girdi verileri, jeton (kelime parçası) yerleştirmeleri gibi ilk katmana beslenir. Her katmanın çıktısı bir sonraki katmanın girdisidir.

The basic idea in deep learning algorithms is stacking up the nonlinear transformation layers. More complex nonlinear transformations can be constructed from deeper layers (Najafabadi, et al., 2015). Through a deep architecture with multiple levels of representations, the data

are transferred into abstract representations. In this case, deep learning algorithms can be considered as a kind of representation learning algorithm.

Derin öğrenme algoritmalarındaki temel fikir, doğrusal olmayan dönüşüm katmanlarını yığmaktır. Daha derin katmanlardan daha karmaşık doğrusal olmayan dönüşümler oluşturulabilir (Najfabadi ve diğerleri, 2015). Birden çok temsil düzeyine sahip derin bir mimari aracılığıyla veriler soyut temsillere aktarılır. Bu durumda derin öğrenme algoritmaları bir nevi temsili öğrenme algoritması olarak düşünülebilir.

The final trained model can be thought of as a highly nonlinear function of the input data which can construct a final representation. The underlying explanatory factors in the data can be extracted from the nonlinear transformations by the layers of deep architecture (Najafabadi, et al., 2015).

Nihai eğitilmiş model, son bir temsil oluşturabilen girdi verilerinin oldukça doğrusal olmayan bir işlevi olarak düşünülebilir. Verilerdeki temel açıklayıcı faktörler, derin mimari katmanları tarafından doğrusal olmayan dönüşümlerden çıkarılabilir (Najfabadi, et al., 2015).

The final representation (the output of the final layer) contains the useful information in the training data, constructed by the deep learning algorithm, which can be used as the features in building classifiers in a high-efficiency comparing with the high dimensional sensory data.

Nihai temsil (son katmanın çıktısı), yüksek boyutlu duyusal verilerle karşılaştırıldığında yüksek verimlilikte sınıflandırıcılar oluşturmada özellikler olarak kullanılabilen derin öğrenme algoritması tarafından oluşturulan eğitim verilerindeki faydalı bilgileri içerir.

In this thesis, we used a pre-trained language representation model built in deep learning techniques to extract the information from text and transfer the text into vectors. The output of the language model called work embeddings containing the information of the input text.

Bu tezde, metinden bilgi çıkarmak ve metni vektörlere aktarmak için derin öğrenme teknikleriyle oluşturulmuş önceden eğitilmiş bir dil temsil modeli kullandık. Dil modelinin çıktısı, girdi metninin bilgilerini içeren çalışma yerleştirmeleri olarak adlandırılır.

## Pre-trained Language Models

Due to the effectiveness of the pre-trained language model in many NLP downstream tasks, it has received much attention.

Birçok NLP alt görevinde önceden eğitilmiş dil modelinin etkinliği nedeniyle, çok dikkat çekmiştir.

Language model pre-training has been proved the efficiency for improving many natural language processing tasks such as sentiment classification (Xipeng, et al., 2020). The basic idea behind the pre-trained language model is training a word embedding layer from a large scale of style so that it has an excellent ability to extract the information from contextual text. Because it is not enough to train various neural architectures of coding context representation only from the limited supervision data of terminal tasks.

Dil modeli ön eğitiminin, duygu sınıflandırması gibi birçok doğal dil işleme görevini geliştirmedeki etkinliği kanıtlanmıştır (Xipeng, et al., 2020). Önceden eğitilmiş dil modelinin arkasındaki temel fikir, bağlamsal metinden bilgi çıkarma konusunda mükemmel bir yeteneğe sahip olması için geniş bir stil ölçeğinden bir kelime gömme katmanı eğitmektir. Çünkü sadece terminal görevlerinin sınırlı denetim verilerinden kodlama bağlamı temsilinin çeşitli sinir mimarilerini eğitmek yeterli değildir.

Bidirectional Encoder Representations from Transformers (BERT) is a pre-trained language representation model proposed based on deep learning techniques by Google AI team in 2018 (Devlin, et al., 2019). Different with other language representation models, through jointly condition on both left and right context in all layers, BERT can generate deep bidirectional representations from the unlabeled input text. BERT has been applied in various NLP tasks such as text classification and question answering and preformed an excellent performance (Yuwen and Zhaozhuo, 2018).

Transformers'dan Çift Yönlü Kodlayıcı Temsilleri (BERT), Google AI ekibi tarafından 2018'de derin öğrenme tekniklerine dayalı olarak önerilen önceden eğitilmiş bir dil temsil modelidir (Devlin ve diğerleri, 2019). Diğer dil temsil modellerinden farklı olarak, tüm katmanlarda hem sol hem de sağ bağlamda ortak koşul aracılığıyla BERT, etiketlenmemiş giriş metninden derin çift yönlü temsiller üretebilir. BERT, metin sınıflandırma ve soru cevaplama gibi çeşitli NLP görevlerinde uygulanmış ve mükemmel bir performans sergilemiştir (Yuwen ve Zhaozhuo, 2018).

Because of the fine-tuning approach adopted, there is no specific architecture for downstream NLP tasks when we use BERT. As an intelligent agent, it should minimize the use of prior human knowledge in the model design and learn such knowledge from data instead of. In BERT, there are two different objectives used to train the language model rather than the frequently used objective of next-word prediction: The first is the masked language model objective, where the model needs to predict the masked tokens from their context. The other one is the next-sequence prediction objective, where the model needs to learn whether sequence B follows sequence A. Those two objectives enable the model to learn long-term dependencies better.

Benimsenen ince ayar yaklaşımı nedeniyle, BERT kullandığımızda aşağı akış NLP görevleri için belirli bir mimari yoktur. Akıllı bir etmen olarak, model tasarımında önceki insan bilgisinin kullanımını en aza indirmeli ve bu bilgileri veri yerine veriden öğrenmelidir. BERT'de, bir sonraki kelime tahmininin sık kullanılan amacından ziyade dil modelini eğitmek için kullanılan iki farklı hedef vardır: Birincisi, maskeli dil modeli hedefidir ve modelin maskelenmiş belirteçleri bağlamlarından tahmin etmesi gerekir. Diğeri, modelin B dizisinin A dizisini takip edip etmediğini öğrenmesi gereken bir sonraki dizi tahmin hedefidir. Bu iki amaç, modelin uzun vadeli bağımlılıkları daha iyi öğrenmesini sağlar

* **Masked Language Model Objective:** The model learns to predict the tokens masked out randomly in sequence A and sequence B.
* Maskeli Dil Modeli Amaç: Model, A dizisi ve B dizisinde rastgele maskelenen belirteçleri tahmin etmeyi öğrenir.
* **Next-Sentence Prediction:** In order to enable BERT to learn long-term dependencies better, the model needs to learn if a sequence B would naturally follow the previous sequence A. So the sequence A and sequence B are from the same document so that sequence A follows sequence B.
* Sonraki Cümle Tahmini: BERT'nin uzun vadeli bağımlılıkları daha iyi öğrenmesini sağlamak için, modelin bir B dizisinin doğal olarak önceki A dizisini izleyip izlemeyeceğini öğrenmesi gerekir. Dolayısıyla A dizisi ve B dizisi aynı belgedendir, böylece A dizisi B dizisini takip eder.

In BERT (Devlin et al., 2019), the authors use the transformer as basic components rather than recurrent or convolutional neural networks. The transformer is solely based on the self- attention mechanism. Compared with Recurrent Neural Network (RNN) or Convolutional Neural Network (CNN), the transformer has three advantages. Firstly, it can reduce the computation resource and computation speed. Secondly, the computation can be parallelized which is impossible in RNN. Otherwise, the transformer has a good performance in learning long-range dependencies.

BERT'de (Devlin ve diğerleri, 2019), yazarlar transformatörü tekrarlayan veya evrişimli sinir ağları yerine temel bileşenler olarak kullanırlar. Transformatör yalnızca kendi kendine dikkat mekanizmasına dayanmaktadır. Tekrarlayan Sinir Ağı (RNN) veya Evrişimli Sinir Ağı (CNN) ile karşılaştırıldığında, transformatörün üç avantajı vardır. İlk olarak, hesaplama kaynağını ve hesaplama hızını azaltabilir. İkinci olarak, RNN'de imkansız olan hesaplama paralelleştirilebilir. Aksi takdirde, transformatör uzun menzilli bağımlılıkları öğrenmede iyi bir performansa sahiptir.

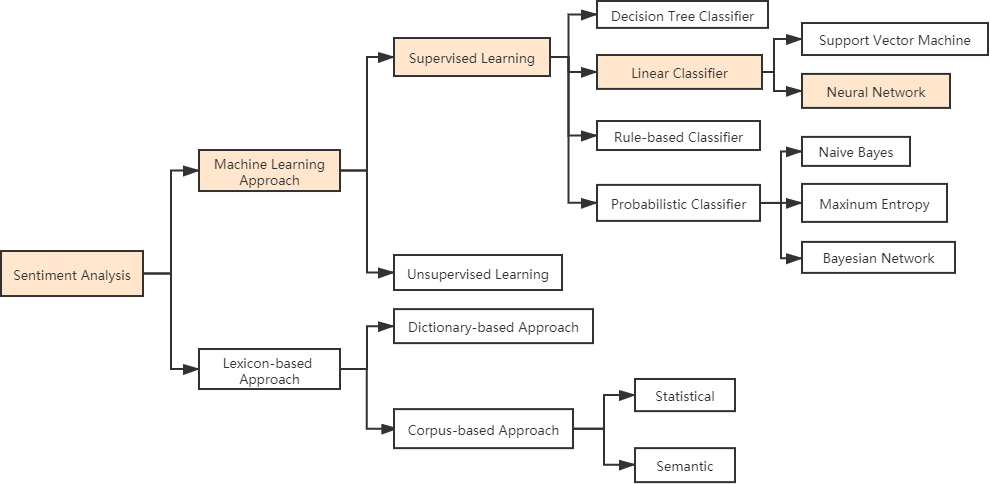
In practice, it is easy to create an excellent performance model by fine-tuning the BERT with one additional output layer for various NLP tasks such as text classification and question answering, without too much substantial task-specific architecture modifications (Devlin et al., 2019).

Pratikte, metin sınıflandırma ve soru yanıtlama gibi çeşitli NLP görevleri için çok fazla önemli göreve özgü mimari değişiklik yapmadan BERT'ye bir ek çıktı katmanıyla ince ayar yaparak mükemmel bir performans modeli oluşturmak kolaydır (Devlin ve diğerleri, 2019).

## Sentiment Analysis (SA)

Sentiment analysis (SA, also known as opinion mining) is defined as a computational task of finding people's opinions about specific entities. There are three main classification levels in sentiment analysis (Medhat and Hassan, 2014): document-level, sentence-level, and aspect-level sentiment analysis. The purpose of document level emotion analysis is to classify opinion documents as expressing positive or negative opinions or emotions. It considers the whole document as a basic unit of information. Sentence-level SA aims to classify sentiment expressed in each sentence. Actually, we can think of sentences as a short document so that there is no fundamental difference between document-level and sentence- level (Liu B., 2012). Aspect-level sentiment analysis will not be discussed in this thesis.

Duygu analizi (SA, aynı zamanda fikir madenciliği olarak da bilinir), belirli varlıklar hakkında insanların fikirlerini bulmaya yönelik hesaplamalı bir görev olarak tanımlanır. Duyarlılık analizinde üç ana sınıflandırma düzeyi vardır (Medhat ve Hassan, 2014): belge düzeyinde, cümle düzeyinde ve en-boy düzeyinde duygu analizi. Belge düzeyinde duygu analizinin amacı, fikir belgelerini olumlu veya olumsuz görüş veya duyguları ifade eden olarak sınıflandırmaktır. Tüm belgeyi temel bir bilgi birimi olarak kabul eder. Cümle düzeyinde SA, her cümlede ifade edilen duyguyu sınıflandırmayı amaçlar. Aslında cümleleri kısa bir belge olarak düşünebiliriz, böylece belge düzeyi ile cümle düzeyi arasında temel bir fark yoktur (Liu B., 2012). En-boy düzeyinde duygu analizi bu tezde tartışılmayacaktır.



**Figure 5.** Sentiment classification techniques (Maynard and Funk, 2012). In this thesis, we use Supervised machine learning with a neural network linear classifier.

Sentiment classification techniques can be divided into two categories: The detailed algorithms are showing in Figure 5. Sentiment Classification techniques can be roughly divided into the hybrid approach, machine learning approach and the lexicon-based approach (Maynard and Funk, 2012). The Machine Learning (ML) approach applies ML algorithms and uses linguistic features. The lexicon-based approach relies on a sentiment lexicon, which is a collection of precompiled and known sentiment terms. More detailed, it can be divided into the dictionary-based approach and corpus-based approach which use statistical and semantic methods to find sentiment polarity, respectively. The hybrid approach combines both approaches played a critical role in the majority of the methods which is very common with sentiment lexicons (Medhat and Hassan, 2014).

Duygu sınıflandırma teknikleri iki kategoriye ayrılabilir: Ayrıntılı algoritmalar Şekil 5'te gösterilmektedir. Duyarlılık Sınıflandırma teknikleri kabaca hibrit yaklaşım, makine öğrenimi yaklaşımı ve sözlük tabanlı yaklaşıma ayrılabilir (Maynard ve Funk, 2012). Makine Öğrenimi (ML) yaklaşımı, makine öğrenimi algoritmalarını uygular ve dilsel özellikleri kullanır. Sözlüğe dayalı yaklaşım, önceden derlenmiş ve bilinen duygu terimlerinin bir koleksiyonu olan bir duygu sözlüğüne dayanır. Daha ayrıntılı olarak, duygu polaritesini bulmak için sırasıyla istatistiksel ve semantik yöntemleri kullanan sözlük tabanlı yaklaşım ve derlem tabanlı yaklaşım olarak ikiye ayrılabilir. Melez yaklaşım, duygu sözlüklerinde çok yaygın olan yöntemlerin çoğunda kritik bir rol oynayan her iki yaklaşımı birleştirir (Medhat ve Hassan, 2014).

The sentiment classification method using lexicon-based approach can be divided into the dictionary-based approach and the corpus-based approach, which depends on finding the sentiment lexicon. The dictionary-based approach begins with finding sentiment or opinion seed words and then searches the dictionary of their synonyms and antonyms. The corpus- based approach depends on a seed list of opinion words and then finds other sentiment words using statistical or semantic methods in a large corpus to help in finding sentiment words with context-specific orientations.

Sözlük tabanlı yaklaşımı kullanan duygu sınıflandırma yöntemi, sözlük tabanlı yaklaşım ve duygu sözlüğünün bulunmasına bağlı olan derlem tabanlı yaklaşım olarak ikiye ayrılabilir. Sözlük temelli yaklaşım, duygu veya fikir tohumları bulmakla başlar ve daha sonra eş anlamlı ve zıt anlamlılarının sözlüğünde arama yapar. Derlem tabanlı yaklaşım, fikir kelimelerinin bir tohum listesine dayanır ve daha sonra, bağlama özel yönelimleri olan duygu kelimelerini bulmaya yardımcı olmak için büyük bir bütünde istatistiksel veya anlamsal yöntemler kullanarak diğer duygu kelimelerini bulur.

Machine learning approaches are the dominant approaches in the sentiment analysis task (Read, 2005). It depends on the features of data when used to sentiment analysis. There are two approaches: unsupervised and supervised learning methods. The supervised methods make use of a large number of labeled training documents. The unsupervised methods are used when it is difficult to find these labeled training documents when they do not exist (Medhat and Hassan, 2014).

Duygu analizi görevinde makine öğrenmesi yaklaşımları baskın yaklaşımlardır (Read, 2005). Duygu analizi için kullanıldığında verilerin özelliklerine bağlıdır. İki yaklaşım vardır: denetimsiz ve denetimli öğrenme yöntemleri. Denetimli yöntemler, çok sayıda etiketli eğitim belgesinden yararlanır. Denetimsiz yöntemler, bu etiketli eğitim belgelerinin bulunmadığı durumlarda bulunmasının zor olduğu durumlarda kullanılmaktadır (Medhat ve Hassan, 2014).

The Bag Of Words (BOW) (Zhang, Jin and Zhou, Z., 2010) model is a traditional ML approach that is frequently used. The main idea is to map feature vectors from a document and then classify by machine learning techniques. Despite the simplicity and efficiency of the BOW method, a lot of the information from the original natural language is lost (Xia and Zong, 2010) because various types of features have been exploited, such as word order and syntactic structures (Pak and Paroubek, 2010).

Bag Of Words (BOW) (Zhang, Jin ve Zhou, Z., 2010) modeli sıklıkla kullanılan geleneksel bir ML yaklaşımıdır. Ana fikir, bir belgeden özellik vektörlerini eşlemek ve ardından makine öğrenimi teknikleriyle sınıflandırmaktır. BOW yönteminin basitliğine ve verimliliğine rağmen, orijinal doğal dildeki birçok bilgi kaybolur (Xia ve Zong, 2010), çünkü kelime sırası ve sözdizimsel yapılar gibi çeşitli özelliklerden yararlanılmıştır (Pak ve Paroubek, 2010).

In general, traditional approaches such as Support Vector Machine (SVM) is based on complex manually extracted feature, which is a time-consuming and complex process (Agarwal, et al., 2011). Traditional machine learning methods contain many steps and fundamental questions like complex features extraction from text data, figuring out the relevant features, and selecting a suitable classification algorithm for the tasks (Sharma and Dey, 2012).

Genel olarak, Destek Vektör Makinesi (SVM) gibi geleneksel yaklaşımlar, zaman alıcı ve karmaşık bir süreç olan karmaşık manuel olarak çıkarılan özniteliğe dayanmaktadır (Agarwal, vd., 2011). Geleneksel makine öğrenme yöntemleri, metin verilerinden karmaşık özelliklerin çıkarılması, ilgili özelliklerin bulunması ve görevler için uygun bir sınıflandırma algoritmasının seçilmesi gibi birçok adım ve temel soruyu içerir (Sharma ve Dey, 2012).

Deep learning is an increasingly popular alternative to traditional machine learning methods because of its excellent performance in Natural Language Processing (NLP) tasks such as sentiment analysis (Collobert et al., 2011). Compared with traditional methods, more complex features can be extracted from the data automatically when using neural networks but with minimum external contribution (Bengio, 2009). Figure 1 shows a clear difference between those different techniques: Compared with traditional methods, deep learning could help to extract features automatically rather manually. While deep learning techniques have been applied to many NLP tasks, usually those models required large datasets and high-performance computational resources for training (Blitzer, Dredze and Pereira, 2007).

Derin öğrenme, duygu analizi gibi Doğal Dil İşleme (NLP) görevlerindeki mükemmel performansı nedeniyle geleneksel makine öğrenimi yöntemlerine giderek daha popüler bir alternatiftir (Collobert ve diğerleri, 2011). Geleneksel yöntemlerle karşılaştırıldığında, yapay sinir ağları kullanılırken, minimum dış katkı ile verilerden otomatik olarak daha karmaşık özellikler çıkarılabilir (Bengio, 2009). Şekil 1, bu farklı teknikler arasındaki açık farkı göstermektedir: Geleneksel yöntemlerle karşılaştırıldığında, derin öğrenme, özelliklerin manuel olarak yerine otomatik olarak çıkarılmasına yardımcı olabilir. Derin öğrenme teknikleri birçok NLP görevine uygulanmış olsa da, genellikle bu modeller eğitim için büyük veri kümeleri ve yüksek performanslı hesaplama kaynakları gerektiriyordu (Blitzer, Dredze ve Pereira, 2007).

In this thesis, BERT deep learning is used to do sentiment analysis, which belongs to neural network, performed as in Figure 5 (Medhat and Hassan, 2014).

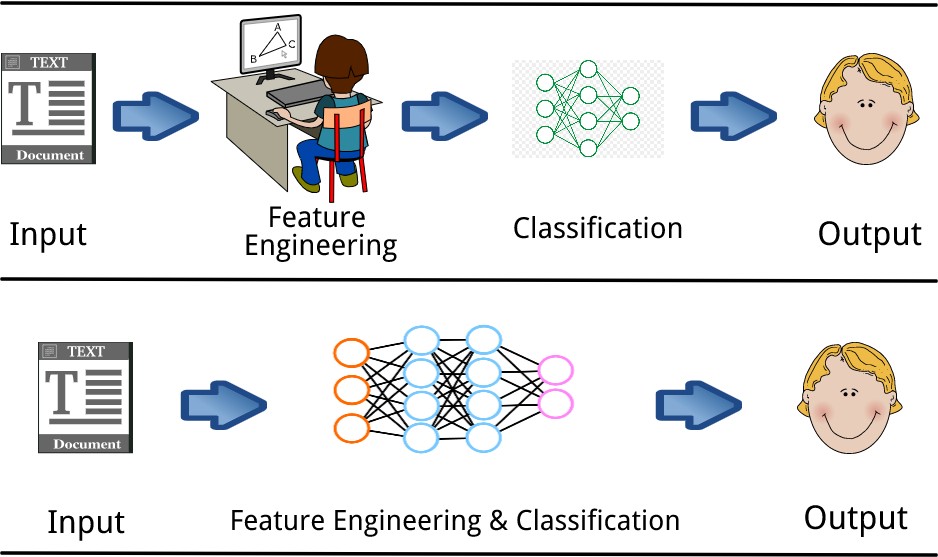
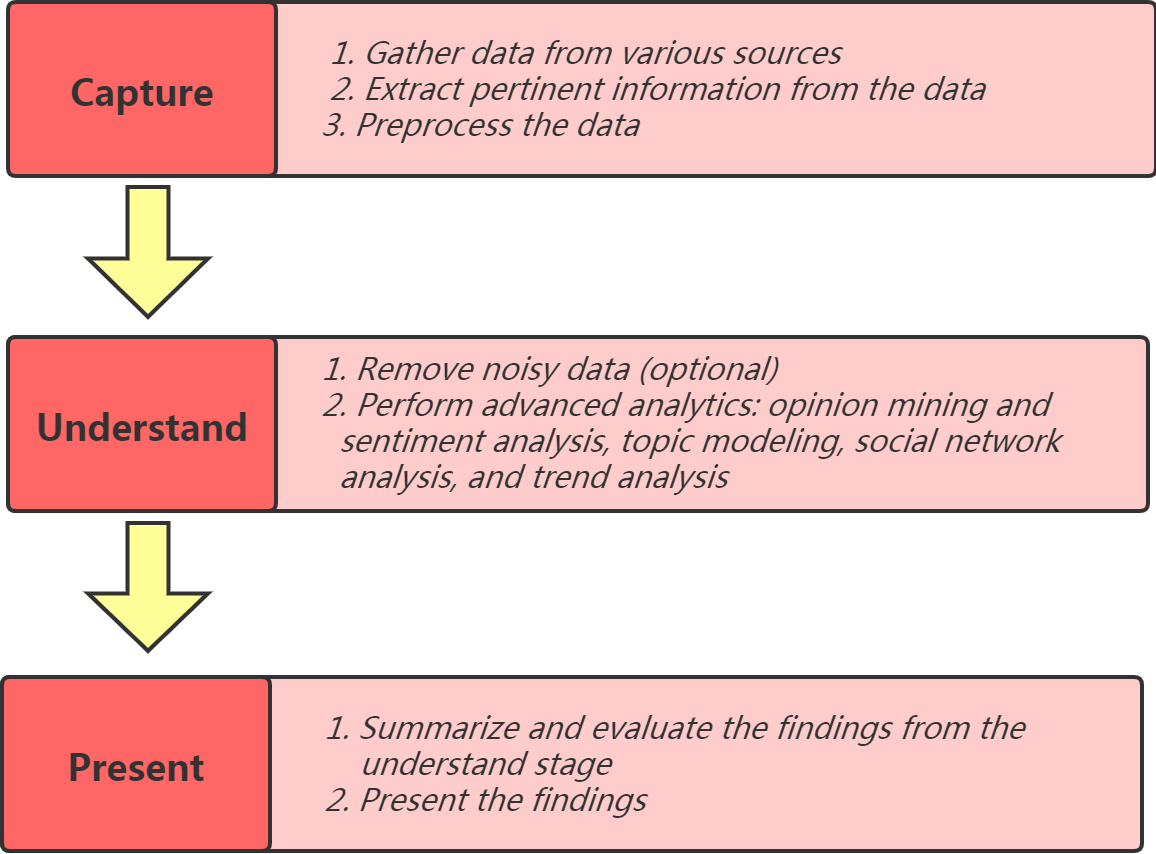


Figure 1. Difference between machine learning and deep learning.

## Social Media Analytics

The Internet and mobile technologies are the main forces of the rise of social media, providing a technical platform for information dissemination, content generation and interactive communication. Social media has become a critical part of the information ecosystem (Daniel, et al., 2010).

Over the past few years, the research of social media has greatly intensified by the significant interest from different domains. Social media analytics, usually driven by specific requirements by a target application, involves developing and evaluating informatics tools and frameworks to collect, monitor, analyze, summarize, and visualize social media data (Daniel, et al., 2010). Social media analytics contains a three-stage process: "capture", "understand" and "present". [Figure 4](#_bookmark11) presents this process with explanations.



**Figure 4**. Social media analytics process (Weiguo and Michael, 2014).

**Capture**: The capture stage is helpful to identify information on social media platforms related to its activities and interests by collecting enormous amounts of relevant data from many social media sources. These data are archived and available to meet the requirements of task. Through various preprocessing steps, including data modeling, data and record linking from different sources, stemming, part-of-speech tagging, feature extraction, and other syntactic and semantic operations that support the analysis, the processed data are delivered to the understanding stage**.**

**Understand**: Usually, there is a considerable part of the noisy data that exists in the data collected from many uses and sources on the capture stage, which need to be removed before meaningful analysis. Then, many techniques from machine translation, text, natural

language processing, data mining and network analysis can be involved in accessing meaning from the cleaned data (Fan and Gordon, 2014).

At this stage, many useful metrics and trends about users can be produced, covering users' backgrounds, interests, concerns and relationship networks. Note that the understanding stage is the heart of the entire social media analysis process. Its results will have a significant impact on the information and metrics in the present stage, these results will be of great help to the decision-making of businesses.

**Present**: As the last stage, the results from different analytics are evaluated, summarized and shown in an easy-to-understand format. Various visualization techniques can be used to present useful information.

In this thesis, the main process of sentiment analysis follows the social media analytics process of [Figure 4.](#_bookmark11) The capture stage corresponds to data description and data preprocessing. The method part will mainly focus on understanding and extracting information for study goals. Lastly, the results will be presented and analyzed in Section 5, which is the ”present” stage.

# Research Methodology

## Literature Review

The main purpose of the literature review is to help researchers to gain knowledge of prior work in the specific field. So researchers can know what has been done well and what needs improvement. There are various sources for reviewing the literature, but not all of them are used here. In this thesis, Dalarna University’s database Summon and Google are used to search related sources, including books, academic journal articles, thesis and websites.

## Research Strategy

In order to exlore the variation of public sentiment during epidemic outbreaking period, a sentiment calssification model is conducted which is a IT product. From Ocate’s book, the research straregy in this thesis is design and creation as a artefact is created.

## Data Generation

According to Oates, the data generation method is document because it already exist prior to the research (Oates, 2005).

The data used in this thesis are Weibo microblog text-based posts published by the Beijing government (Datafountain, 2020). There are 1 million posts, with 10% labeled data, and 90% unlabeled data.

## Data Analysis

Qualitative data includes all non-numeric data – words, images, sounds, and so on – found in such things as interview tapes, researchers' diaries, company documents, websites, and developers' models (Oates, 2005). So the data analysis method can also call qualitative data analysis.

In this thesis, the process of data analysis is followed social media analytics in section 2.4. The detailed experiment process are combined used BERT deep learning as our data analysis method and combined with social media analytics.

## Experiment Process

From the above idea, in this work, BERT is applied to sentient analysis. It helps to extract the information from the text and transform the text into vectors. Because of its excellent performance and easy migration to other tasks, it has led a tendency to applied transfer learning into the NLP domain. Based on this, a sentiment classifier will be developed for public sentiment variation based on microblog data. The focus of this thesis is on studying

how the epidemic affects public sentiments instead of improving the sentiment analysis method. [Figure 2](#_bookmark18) shows an overview of the experiment process in thesis from start to end.

According to social media analytics in Section 2.4, the experiment process can be divided into 3 stages:

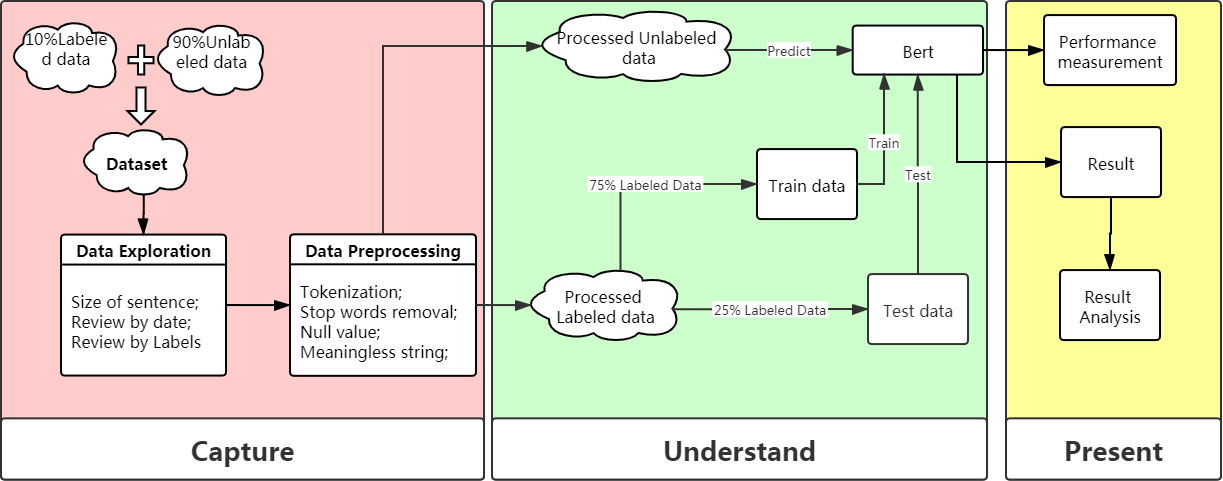
## Capture stage:

* Gather data: The dataset is from Beijing government including 100,000 labeled and 900,000 unlabeled blog posts. According to
* Extract pertinent information: which is corresponding with data exploration, helps to understand what is in a dataset and the characteristics of the data visual exploration. A set of methodologies are used to discover and evaluate appropriate problems.
* Preprocess data: In this thesis, data preprocessing includes tokenization, cleaning, and transformation of data. The result expected after reliable chaining of data preprocessing tasks is a final dataset, which can be considered correct and useful for further data analysis algorithms (García, Luengo and Herrera, 2014). Both labeled data and unlabeled data need to be processed so that processed labeled data and processed unlabeled are output.

## Understand stage:

* Train Phase: Processed labeled data will be split into training data and test data uniformly at random. 75% of the labeled data are used to train sentiment classification model.
* Test Phase: The rest 25% of the labeled data are used to validate the performance of the model.
* Predict Phase: Processed unlabeled data will be annotated sentiment labels by the trained model.

**Present stage:** Finally, both labeled data and all data (unlabeled data and labeled data) are used for public sentiment variation analysis. The result analysis will be present at this stage.



**Figure 2**. Experiment process.

# Data

## Data Exploration

According to the information from the government, the sentiments in labeled data for each post are annotated manually so that the accuracy of labeled data can be guaranteed.

The length of microblogs is important for model development as it decides the length of the model input. Next, we present an overview distribution of the length of al posts. Before showing its distribution, kernel density estimates will be introduced first. Kernel density estimates (KDEs), a technique can convert continuous data such as histograms or scatter plots into a smoothed probability density function in a nonparametric way with a smoothing parameter h, called the bandwidth. A kernel density estimate of a univariate probability density function (pdf) *f* based on a random sample 𝑥𝑖, 𝑖 = 1, … , 𝑛 (Thomas H., Christoph

W., 2007) is

𝑓̂(𝑥) = 1 𝛴𝑛

𝑥−𝑥𝑖)*,*

𝑛ℎ

𝑖=1𝐾 ( ℎ

where n is the sample size, *h* is the kernel bandwidth and the kernel *K* is a kernel function. The bandwidth *h* determines the smoothness of the estimated density. A smaller bandwidth h will produce an estimated density with higher variability while a greater *h* leads a smoother estimated density. Kernel functions are able to estimate the density of random variables. The kernel function used here is Gaussian kernel function, defined as (Che and Wang, 2014)

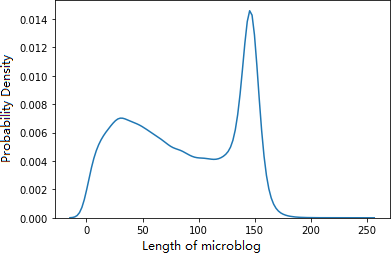
𝐾(𝑥) = 1 𝑒

𝜎√2𝜋

−(𝑥−𝜇)2

2𝜎2 .

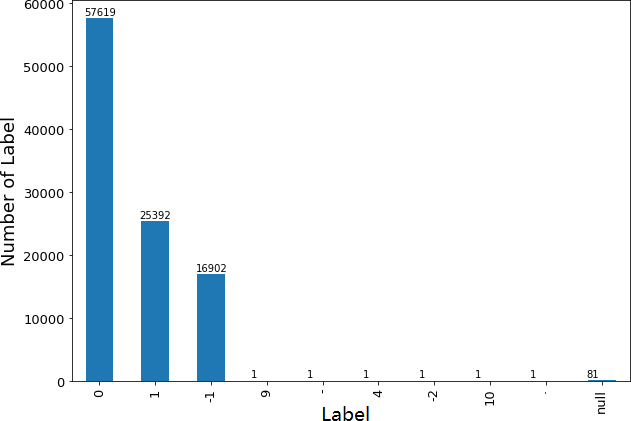
When implementing KDE graph, the high-level interface is provided for drawing KDE graphics by Seaborn which is a Python data visualization library based on Matplotlib. [Figure](#_bookmark22) [6](#_bookmark22) shows the distribution of the number of characters; each post contains no more than 260 characters. Mainly microblogs have 30 to 180 characters.



**Figure** **6.** KDE estimate of the number of characters.

[Figure 7](#_bookmark23) shows the number of each sentiment label: -1 represents negative, 0 represents neutral and 1 represents positive. It is obvious that outstrip 60% of blog posts are neutral,

approximately 17,000 blogs are labeled as negative. At the same time, 87 of the labels are noise, shown in Figure 7, where the x-axis is the label and y-axis is the number of each sentiment label.



**Figure** **7.** The number of labels.

## Data Preprocessing

[Figure 8](#_bookmark25) illustrates the process of data preprocessing step by step. Microblog data includes labeled data and unlabeled data, data preprocessing is essential for both of them. The detailed explanation is shown as follows.

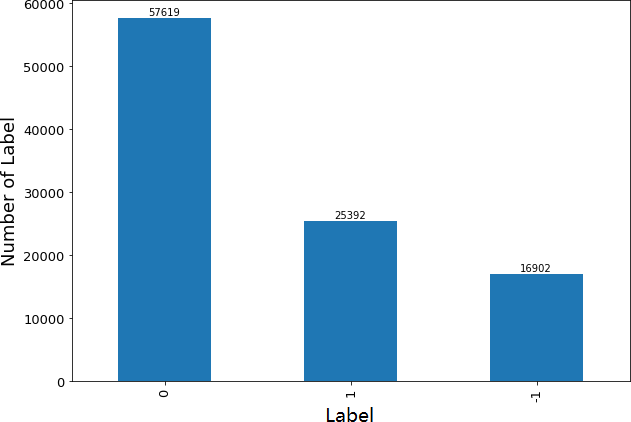


**Figure 8.** The flowchart of data preprocessing.

1. **Remove Meaningless Field:** Some specific fields need to be removed including the @mention, URL, punctuation, hashtag and white space, which do not provide any relevant information about the text sentiment. Any URLs starting “HTTP”, “https” from the blog posts will be removed. Whitespace does not provide any meaning to the text, so it is removed.
2. **Segmentation**: The word is the smallest independent meaningful element in Chinese which is different with English consist of the character. In the Chinese word segmentation, there is no obvious boundary between the words when doing Chinese lexical analysis. Such word segmentation, a very challenging problem because it is hard to define what constitutes a word (Jianfeng, et al., 2005). Although some criteria are

helpful to do it such as grammar rule, they do not consistently lead us to the same conclusion. Otherwise, no white space exists between Chinese words or expressions and there are many ambiguities in the Chinese language. Also, word-stemming is not applicable because of no obvious inflected or derived words in Chinese. Therefore, to reduce the noise brought by Chinese word segmentation to get a better word list for one document or sentence, word segmentation is applied to the Chinese text.

1. **Remove Stopwords**: Stopwords are meaningless for the understanding of public sentiment. Stopwords are the words that appear in the texts frequently but do not carry significant information. From the result of statistical analysis through documents, some words have quite low frequency usually act just the opposite (Feng, et al., 2006). For instance, the words "the”, “and”, “of” appear frequently in the English text. Those words are used just because of grammar while no significant information exists for text understanding stage (Feng, et al., 2006). The Chinese stop word list used here is from Harbin industrial university.
2. **Remove Items with Noise Label:** Noised labels are an unavoidable problem. The quality of the training data is a decisive factor to the performance of the model. The quality of the class labels refers to whether the class of each example is assigned correctly (García, Luengo and Herrera, 2014). In this study, the simplest strategy is used as there are only 279 blogs with incorrect labels, occupying a small part in all labeled data only 0.279%. We simply remove those blogs directly as they have little influence on the result. The distribution of labels after processing is shown in [Figure 9.](#_bookmark26)



**Figure 9**. The distribution of sentiments for labeled data.

1. **Split Dataset:** To increase confidence in the prediction ability of a model, validation is performed. Split sample validation consists of splitting available data into two samples. One sample is used to train the model and the other one to test the prediction ability of

the model. In this thesis, labeled microblog data are split uniformly at random into 25% test data and 75% train data.

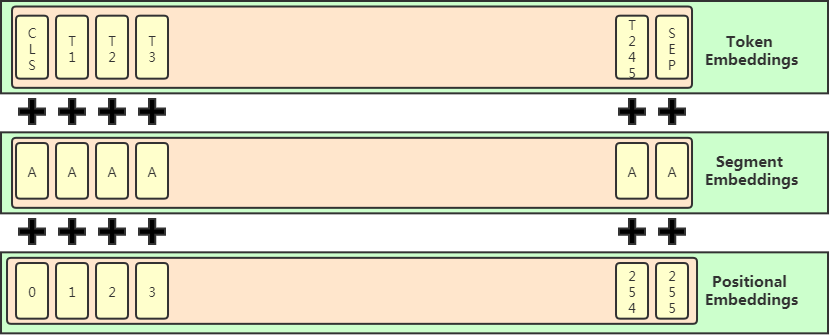
1. **Initial Input Embedding:** The processed data will be transferred into an input vector. As in (Devlin et al., 2019), we tokenize every micro-blog using a 30,522 word-piece vocabulary contains frequent Chinese characters and special tokens (like [CLS] and [SEP]), then generate multiple input instances per example by concatenating a [CLS] token, tokens from the content of the micro-blogs, and a final “[SEP]” token, limiting the total size of each instance to 512 tokens. Here input size will be set as 256 tokens as the blog posts are less than 260 tokens. Given a blog posts 𝑆 = (𝑆1, . . . , 𝑆𝑚), we formulate the input as a sequence X = ([CLS], S1, . . . , S𝑚, [SEP]), where [CLS] is a dummy token not used and [SEP] is intended to separate difference sentence. Now, all

micro-blogs have the common format style, starting with [CLS] tokens, concatenating with tokens of micro-blog and finally ending with [SEP]. After that, tokens need to transfer into initial word embedding as input of model. The input representation is the sum of token embedding, the segment embedding, and the position embedding, different embedding contains different information of the text.

* + **Token Embedding:** Sequence X will be transferred into 𝑡𝑜𝑘𝑒𝑛\_𝑖𝑑𝑠 which is the index of token in vocabulary. For example, the token embedding of X can be represented as 𝑇𝑜𝑘𝑒𝑛𝐸𝑚𝑏𝑒𝑑𝑑𝑖𝑛𝑔 = (𝐼𝑛𝑑𝑒𝑥[𝐶𝐿𝑆],𝐼𝑛𝑑𝑒𝑥𝑆1 , … , 𝐼𝑛𝑑𝑒𝑥[𝑆𝐸𝑃]) , the length of which is the same as that of sequence X. When the size of token embedding *H* is less than fixed input size 256, the token embedding will be

supplemented with (256 − 𝐻) zeros.

* + **Segment Embedding:** In this case, one blog will be seen as an entire sentence. As a result, [SEP] can think of as the separate signal of blog and 𝑠𝑒𝑔𝑚𝑒𝑛𝑡\_𝑖𝑑𝑠 as one blog will be 1 always.
  + **Position Embedding:** In the original text, a magic rule is used to positional embedding. Sine and cosine functions are used to embed the position, this method can extend the position embedding to the sequence advantage of invisible length (for example, inferred) The sequence appearing in the process is longer than any text in the training sample). The input representation can be performed as in [Figure](#_bookmark27) [10.](#_bookmark27)



**Figure 10**. The input representation (Devlin et al., 2019).

# Method

## Model Architecture

Google polished the pre-trained BERT on Github with different sizes. As a base for our experiments, BERT-base is used which has been pre-trained by Google research team with 12 layers and 768 hidden dimensions per token. Totally 110 million parameters it has. The architecture of the model is presented in [Figure 11](#_bookmark31) with the following three parts.

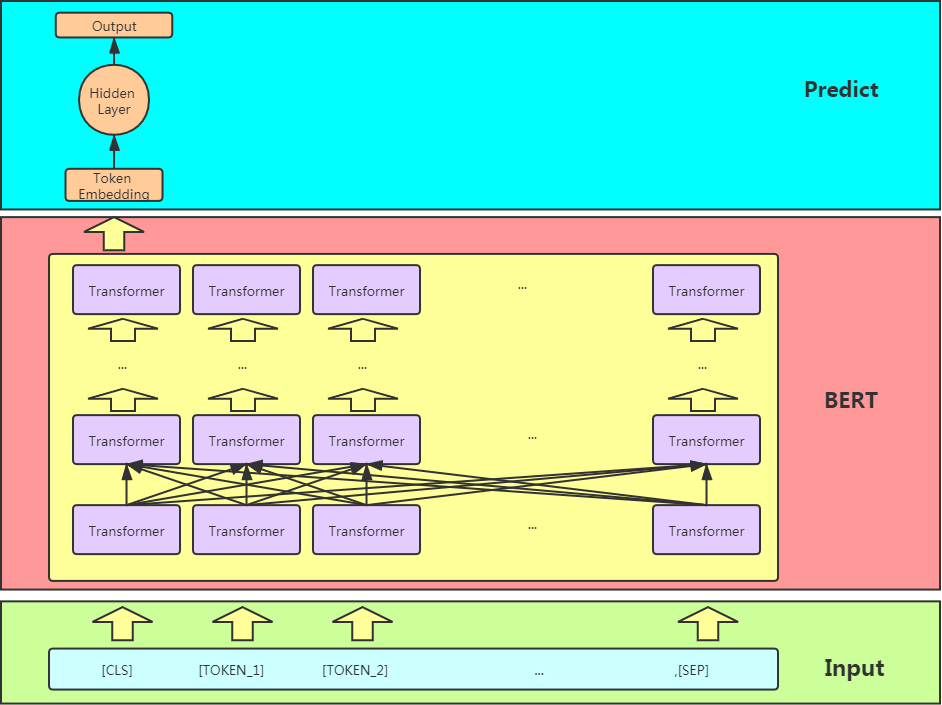
Google, Github'da önceden eğitilmiş BERT'yi farklı boyutlarda cilaladı. Deneylerimiz için bir temel olarak, Google araştırma ekibi tarafından önceden eğitilmiş, 12 katman ve belirteç başına 768 gizli boyut ile BERT tabanı kullanılmıştır. Toplam 110 milyon parametreye sahiptir. Modelin mimarisi Şekil 11'de aşağıdaki üç parça ile sunulmaktadır.

* **Input:** The first part is the input layer. Its responsibility is accepting the initial word embedding and delivering it into BERT. About how to do initial word embedding has been described in [Section 3.2.](#_bookmark24)
* Giriş: İlk kısım giriş katmanıdır. Sorumluluğu, ilk gömme kelimesini kabul etmek ve onu BERT'e teslim etmektir. İlk sözcük yerleştirmenin nasıl yapılacağı Bölüm 3.2'de açıklanmıştır.
* **BERT:** Let 𝐵𝐸𝑅𝑇(·) be the pre-trained BERT model, *x* be the initial embedding from the input part. We first obtain the hidden representation as 𝑍 = 𝐵𝐸𝑅𝑇(𝑥) ∈ 𝑅𝑟ℎ∗|𝑥|, where |𝑥| is the length of the input sequence and 𝑟ℎ is the size of the hidden dimension. The output of this part is the final word embedding of each input tokens.
* BERT: 𝐵𝐸𝑅𝑇(·) önceden eğitilmiş BERT modeli olsun, x giriş kısmından ilk gömme olsun. Önce gizli gösterimi 𝑍 = 𝐵𝐸𝑅𝑇(𝑥) ∈ 𝑅𝑟ℎ∗|𝑥| olarak elde ederiz, burada |𝑥| giriş dizisinin uzunluğu ve 𝑟ℎ gizli boyutun boyutudur. Bu bölümün çıktısı, her bir girdi belirtecinin son sözcük gömmesidir.
* **Predict:** The hidden representation is passed to a dense layer followed by 𝑠𝑜𝑓𝑡𝑚𝑎𝑥

functions: 𝑔 = 𝑠𝑜𝑓𝑡𝑚𝑎𝑥(𝑤 · 𝑍 + 𝑏) = 𝑠𝑜𝑓𝑡𝑚𝑎𝑥(𝐵𝐸𝑅𝑇(𝑥)) , where 𝑊 ∈

𝑅𝑟ℎ 𝑎𝑛𝑑 𝑏 ∈ 𝑅 . The 𝑠𝑜𝑓𝑡𝑚𝑎𝑥 is applied along the dimension of the sequence. In mathematics, the softmax is a function that takes as input a vector of K real numbers and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. The output is the probabilities of each label: 𝑝𝑟𝑒𝑑𝑖𝑐𝑡\_𝑙𝑎𝑏𝑒𝑙 = 𝑎𝑟𝑔 𝑚𝑎𝑥(𝑔), where output ∈ R𝑘 and 𝑘 equals 3 as three sentiments have in the dataset, including positive, negative, and neutral.

 Tahmin: Gizli gösterim yoğun bir katmana ve ardından 𝑠𝑜𝑓𝑡𝑚𝑎𝑥 fonksiyonlarına iletilir: 𝑔 = 𝑠𝑜𝑓𝑡𝑚𝑎𝑥(𝑤 · 𝑍 + 𝑏) = 𝑠𝑜𝑓𝑡𝑚𝑎𝑥(𝐵𝐸𝑅𝑇(𝑥)) , burada 𝑊 ∈ 𝑅𝑟ℎ 𝑎𝑛𝑑 𝑏 ∈ 𝑅 . 𝑠𝑜𝑓𝑡𝑚𝑎𝑥, dizinin boyutu boyunca uygulanır. Matematikte, softmax, girdi olarak K reel sayı vektörünü alan ve onu girdi sayılarının üstelleriyle orantılı K olasılıklarından oluşan bir olasılık dağılımına normalleştiren bir fonksiyondur. Çıktı, her bir etiketin olasılıklarıdır: 𝑝𝑟𝑒𝑑𝑖𝑐𝑡\_𝑙𝑎𝑏𝑒𝑙 = 𝑎𝑟𝑔 𝑚𝑎𝑥(𝑔), burada çıktı ∈ R𝑘 ve 𝑘, veri kümesinde pozitif, negatif ve nötr dahil olmak üzer e üç duygu olduğu için 3'e eşittir.



**Figure 11**. The architecture of BERT used in sentiment classification.

When training model, the weight adjust by the loss function and create a fitting BERT model by minimizing the loss function. During forward propagation, the outputs of model are the probabilities of possible labels. These probabilities are compared to the target labels. Then, the loss function calculates a penalty for any deviation between the target label and the outputs of model. During backpropagation, the trainable weights are adjusted by calculating the partial derivative of the loss function for each weight. Under normal conditions, a model with lower loss is produced after training on the dataset.

Modeli eğitirken, ağırlık kayıp fonksiyonu ile ayarlanır ve kayıp fonksiyonunu en aza indirerek uygun bir BERT modeli oluşturulur. İleri yayılım sırasında, modelin çıktıları olası etiketlerin olasılıklarıdır. Bu olasılıklar hedef etiketlerle karşılaştırılır. Ardından, kayıp fonksiyonu, hedef etiket ile modelin çıktıları arasındaki herhangi bir sapma için bir ceza hesaplar. Geri yayılım sırasında, eğitilebilir ağırlıklar, her ağırlık için kayıp fonksiyonunun kısmi türevi hesaplanarak ayarlanır. Normal şartlar altında, veri seti üzerinde eğitimden sonra daha düşük kayıplı bir model üretilir.

In this case, the sentiment classification is a function 𝑔: ℝ256 → ℝ𝑀 that maps the input feature space to the label space. Training this model involves minimizing the loss *L* that is defined as the average cross-entropy

Bu durumda, duyarlılık sınıflandırması, girdi özellik alanını etiket alanına eşleyen bir 𝑔: ℝ256 → ℝ𝑀 işlevidir. Bu modeli eğitmek, ortalama çapraz entropi olarak tanımlanan L kaybının en aza indirilmesini içerir.

𝐿 = − 1 ∑𝑁

∑𝑀

𝑦 log 𝑔

(𝑥

; 𝜃) ,

where

𝑁 𝑛=1

𝑚=1

𝑛𝑚

𝑚 𝑛

* + *N* is the number of training examples
  + *M* is the number of classes
  + *Θ* is the classifier parameter set
  + 𝑦𝑛𝑚 is the 𝑚’𝑡ℎ element of one-hot encoded label of sample 𝑥𝑛
  + 𝑓𝑚 is the 𝑚’𝑡ℎ element of *f*
  + 𝑦̂𝑚 is the estimate of training example *n* for class *m.*

𝑛

* N, eğitim örneklerinin sayısıdır*

* M sınıf sayısıdır*

* Θ sınıflandırıcı parametre setidir*

* 𝑦𝑛𝑚, numunenin tek sıcak kodlanmış etiketinin 𝑚’𝑡ℎ öğesidir 𝑥𝑛*

* 𝑓𝑚 f'nin 𝑚'𝑡ℎ öğesidir*

* 𝑦̂𝑚, m sınıfı için eğitim örneği n'nin tahminidir.*

In the perfect case, if the output of the classifier is equal to the real target, then it is completely accurate for the training example and the loss is zero*.*

*Mükemmel durumda, sınıflandırıcının çıktısı gerçek hedefe eşitse, eğitim örneği için tamamen doğrudur ve kayıp sıfırdır.*

## Implementation

For fine-tuning the BERT language model on a specific domain, we use the weights of BERTBASE as a starting point. The fine-tuning model was created on Tensorflow Hub (TFHub) (Mehta, 2019).

Belirli bir etki alanında BERT dil modeline ince ayar yapmak için başlangıç noktası olarak BERTBASE ağırlıklarını kullanırız. İnce ayar modeli Tensorflow Hub'da (TFHub) oluşturuldu (Mehta, 2019).

TFHub is one way to try and test machine learning models (Mehta, 2019). It allows people to reuse the machine learning model. With TFHub, models and modules are treated as analogs to binaries and libraries. With this, it becomes well-defined in terms of importing modules and using functionality from within it.

TFHub, makine öğrenimi modellerini denemenin ve test etmenin bir yoludur (Mehta, 2019). İnsanların makine öğrenimi modelini yeniden kullanmasını sağlar. TFHub ile modeller ve modüller, ikili dosyalar ve kitaplıkların analogları olarak değerlendirilir. Bununla, modülleri içe aktarma ve içindeki işlevselliği kullanma açısından iyi tanımlanmış hale gelir.

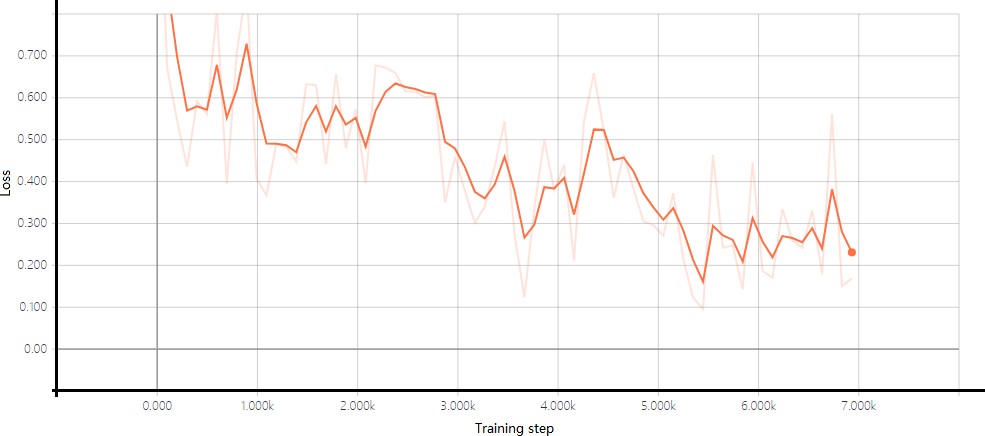
The fine-tuning experiments are based on TFHub. Detailed steps are described as follows.

**Train Phase**: Firstly, we initialize our model from a BERT model already trained on massive Chinese corpus. Then we fine-tuned the model on the training instances precomputed as described in [Section 4.2.](#_bookmark30) We train the model by minimizing loss [*L*](#_bookmark32) with the Adam optimizer (Diederik and Jimmy, 2014) with a batch size of 32. As is common practice for BERT models, we only set the number of epochs as 3 and initial learning rate as 2 ∙ 10−5. For each epoch, there are almost 7,000 blogs, so there are almost 7,000 steps for training. [Figure 12](#_bookmark34) shows the progress of the optimization of the model during the training stage with a smoothing parameter value 0.6.

Tren Aşaması: İlk olarak, modelimizi zaten devasa Çin külliyatında eğitilmiş bir BERT modelinden başlatıyoruz. Ardından, Bölüm 4.2'de açıklandığı gibi önceden hesaplanan eğitim örnekleri üzerinde modele ince ayar yaptık. Parti boyutu 32 olan Adam optimizer (Diederik ve Jimmy, 2014) ile kayıp L'yi en aza indirerek modeli eğitiyoruz. BERT modelleri için yaygın uygulama olduğu gibi, yalnızca dönem sayısını 3 ve ilk öğrenme oranını 2 olarak belirledik ∙ 10−5. Her dönem için yaklaşık 7.000 blog vardır, bu nedenle eğitim için yaklaşık 7.000 adım vardır. Şekil 12, 0,6 yumuşatma parametresi değeriyle eğitim aşaması sırasında modelin optimizasyonunun ilerlemesini gösterir.

**Test Phase:** In order to measure the reliability of the model, performance measurement is an essential stage. 25% of the labeled data are used as test dataset. From the fine-tuned model, each post will be assigned a sentiment label. Later, several measure strategies in [Section 4.4](#_bookmark35) were applied for model performance measurement.

Test Aşaması: Modelin güvenilirliğini ölçmek için performans ölçümü önemli bir aşamadır. Etiketlenen verilerin %25'i test veri seti olarak kullanılır. İnce ayarlanmış modelden, her gönderiye bir duygu etiketi atanacaktır. Daha sonra, model performans ölçümü için Bölüm 4.4'teki çeşitli ölçüm stratejileri uygulandı.



**Figure 12**. The loss graph as a function of the training step.

**Prediction phase**: After fine-tuning the model, unlabeled data are feed into the model to get a prediction for the sentiment. Those data also need to experience data preprocessing step which is described in [Section 3.2](#_bookmark24). Then, all of the unlabeled data will be assigned a sentiment label from the fine-tuned model.

Tahmin aşaması: Modelde ince ayar yapıldıktan sonra, duygu için bir tahmin elde etmek için etiketlenmemiş veriler modele beslenir. Bu verilerin ayrıca Bölüm 3.2'de açıklanan veri ön işleme adımını deneyimlemesi gerekir. Ardından, etiketlenmemiş tüm verilere ince ayarlı modelden bir duygu etiketi atanacaktır.

## Performance Metrics

The input is classified into one, and only one, of *l* non-overlapping class. As for the binary case, multi-class categorization can be objective or subjective, well-defined or ambiguous.

The results of the classification can be described in a confusion matrix ([Table 1](#_bookmark36)) (Sokolova and Lapalme, 2009). Each element 𝐶𝑖,𝑗 describes the number of instances that were predicted as class 𝑖 but belonged to class j. For a multiclass problem with k classes, the confusion matrix is of size 𝑁𝑘∗𝑘. The sum of all elements in the confusion matrix is the

total number of samples 𝑁 presented to the classifier. From the confusion matrix, the number of true positive predictions for each class m is

𝑡𝑝𝑚 = 𝐶𝑚,𝑚.

The number of false negatives for class m is

𝑘

𝑓𝑛𝑚 = ∑

𝑖=1,𝑖≠𝑚

𝐶𝑖,𝑚.

The number of true negative predictions regarding class m can be calculated as

𝑘

𝑡𝑛𝑚 = ∑

𝑖=1,𝑖≠𝑚

𝑘

𝑗=1,𝑗≠𝑚

∑

𝐶𝑖,𝑗.

The true negative predictions regarding class m can be further split into the true negative samples predicted as a sample from each other class n (𝑚 ≠ 𝑛) as

𝑘

𝑡𝑛𝑚 = ∑

𝑛=1,𝑖≠𝑚

𝑡𝑛𝑚,𝑛.

Hence, 𝑡𝑛𝑚,𝑛 is the number of samples that truly did not belong to class m and were predicted as belonging to class n.

Finally, the number of false-positive predictions regarding class 𝑚 are given by

𝑘

𝑓𝑝𝑚 = ∑

𝑖=1,𝑖≠𝑚

𝐶𝑚,𝑖.

Similarly to the true negative predictions, the false positive predictions can be subdivided according to the predicted class 𝑛 (𝑚 = 𝑛)

𝑓𝑝𝑚,𝑛 = 𝐶𝑚,𝑛

such that

𝑘

𝑓𝑝𝑚 = ∑

𝑛=1,𝑛≠𝑚

𝑓𝑝𝑚,𝑛,

which means that 𝑓𝑝𝑚,𝑛 is the number of samples that were predicted as belonging to class m but are truly members of class 𝑛.

**Table** **1:** Confusion matrix for multi-class.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **true** | | | |
| **prediction** | 𝐶1,1 | 𝐶1,2 | … | 𝐶1,𝑘 |
| 𝐶2,1 | 𝐶2,2 | … | 𝐶2,𝑘 |
| … | … | … | … |
| 𝐶𝑘,1 | 𝐶𝑘,2 | … | 𝐶𝑘,𝑘 |

**Table** **2:** Confusion matrix for individual class.

|  |  |
| --- | --- |
| 𝐶𝑙𝑎𝑠𝑠𝑖 | |
| 𝑡𝑝𝑖 | 𝑓𝑝𝑖 |
| 𝑡𝑛𝑖 | 𝑓𝑛𝑖 |

[Table 3](#_bookmark38) presents the metrics for multi-class classification. For an individual class 𝐶𝑖, the assessment is defined by 𝑡𝑝𝑖; 𝑓𝑛𝑖; 𝑡𝑛𝑖; 𝑓𝑝𝑖 which are presented in [Table 2](#_bookmark37).

𝐴𝑐𝑐𝑢𝑟𝑎𝑐𝑦𝑖; 𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛𝑖; 𝑅𝑒𝑐𝑎𝑙𝑙𝑖 are calculated from the counts for 𝐶𝑖 . Quality of the overall classification is usually assessed in two ways: a measure is the average of the same measures calculated for 𝐶1, … , 𝐶𝑙 (macro-averaging is shown with an M index), or the sum of counts to obtain cumulative 𝑡𝑝𝑖; 𝑓𝑛𝑖; 𝑡𝑛𝑖; 𝑓𝑝𝑖 and then calculating a performance

measure (micro-averaging shown with *l* indices). Macro-averaging treats all classes equally while micro-averaging favors bigger classes.

**Table** **3:** Metrics for multi-class classification (Sokolova and Lapalme, 2009).

𝒂

|  |  |  |
| --- | --- | --- |
| **Metrics** | **Formula** | **Evaluation focus** |
| 𝑨𝒗𝒆𝒓𝒂𝒈𝒆 𝑨𝒄𝒄𝒖𝒓 | ∑𝑙 𝑡𝑝𝑖 + 𝑡𝑛𝑖  𝑖=1 𝑡𝑝𝑖 + 𝑓𝑛𝑖 + 𝑓𝑝𝑖 + 𝑡𝑛𝑖  𝑙 | The average per-class effectiveness of a classifier. |
| 𝑷𝒓𝒆𝒄𝒊𝒔𝒊𝒐𝒏𝝁 | ∑𝑙 𝑡𝑝𝑖  𝑖=1  ∑𝑙 𝑡𝑝𝑖 + 𝑓𝑝𝑖  𝑖=1 | Agreement of the data class labels with those of classifiers if calculated from sums of per-text decisions. |
| 𝑹𝒆𝒄𝒂𝒍𝒍𝝁 | ∑𝑙 𝑡𝑝𝑖  𝑖=1  ∑𝑙 (𝑡𝑝𝑖 + 𝑓𝑛𝑖)  𝑖=1 | Effectiveness of a classifier to identify class labels if calculated from sums of per-text decisions. |
| 𝑭𝟏 − 𝒔𝒄𝒐𝒓𝒆𝝁 | (𝛽2 + 1)𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛𝜇𝑅𝑒𝑐𝑎𝑙𝑙𝜇  𝛽2𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛𝜇 + 𝑅𝑒𝑐𝑎𝑙𝑙𝜇 | Relations between data’s positive labels and those given by a classifier based on sums of per-text decisions. |
| 𝑷𝒓𝒆𝒄𝒊𝒔𝒊𝒐𝒏𝑴 | ∑𝑙 𝑡𝑝𝑖  𝑖=1 𝑡𝑝𝑖 + 𝑓𝑝𝑖  𝑙 | An average per-class agreement of the data class labels with those of classifiers. |
| 𝑹𝒆𝒄𝒂𝒍𝒍𝑴 | ∑𝑙 𝑡𝑝𝑖  𝑖=1 𝑡𝑝𝑖 + 𝑓𝑝𝑖  𝑙 | An average per-class effectiveness of a classifier to identify class labels. |
| 𝑭𝟏 − 𝒔𝒄𝒐𝒓𝒆𝑴 | (𝛽2 + 1)𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛𝑀𝑅𝑒𝑐𝑎𝑙𝑙  𝛽2𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛𝑀 + 𝑅𝑒𝑐𝑎𝑙𝑙𝑀 | Relations between data’s positive labels and those given by a classifier based on a per-class average. |

25% of the labeled data are used to test the performance of the model, i.e., roughly 25,000 microblogs. The data preprocessing and input representation are the same as the strategy described in [Section 3.2](#_bookmark24). [Table 4](#_bookmark39) shows the result of the confusion matrix for multiclass which corresponds to Table 1 while Table 5 corresponds to Table 2.

**Table** **4:** Result of confusion matrix for sentiment classification.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **true** | | |
|  | **Label** | **Negative** | **Neutral** | **Positive** |
| **prediction** | **Negative** | 2650 | 1047 | 126 |
| **Neutral** | 1299 | 11635 | 1460 |
| **Positive** | 162 | 1736 | 4416 |

**Table** **5:** Result of confusion matrix for each class or sample.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Class** | **Positive** | | **Neutral** | | **Negative** | |
| **Confusion Matrix** | 19247 | 1461 | 7354 | 3143 | 16991 | 1586 |
| 1533 | 2650 | 2759 | 11635 | 1898 | 4416 |

From [Table 5,](#_bookmark40) it is easy to compute the performance metrics according to the formulas shown in [Table 3](#_bookmark38) with each different metrics. The result of the performance metrics is shown at [Table 6,](#_bookmark41) which can be the answer of question 3. Seven measurement strategies were applied. The values of different strategies were located between 0.71 to 0.76 without a big difference. The model has a balanced performance on the dataset with different labels.

**Table** **6:** The numerical values of each metric.

|  |  |
| --- | --- |
| 𝑴𝒆𝒕𝒓𝒊𝒄 | 𝑽𝒂𝒍𝒖𝒆 |
| accuracy | 0.7513 |
| f1\_score\_micro | 0.7513 |
| f1\_score\_macro | 0.7179 |
| precision\_micro | 0.7513 |
| Precision\_macro | 0.7226 |
| Recall\_micro | 0.7317 |
| Recall\_macro | 0.7513 |

# Results

*Two different visualization strategies are used to show sentiment variation: sentiment KDEs and waterfall charts.*

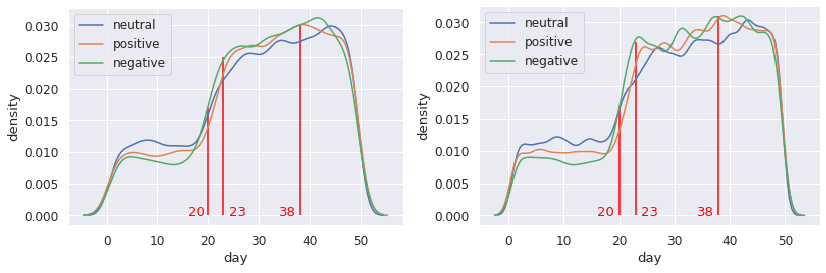
## Sentiment KDEs

KDE plot based on labeled data is presented in [Figure 13 (a)](#_bookmark44) which provides an estimated distribution of each sentiment. This KDE plot is compared with the KDE plot based on labeled data in [Figure 13 (b).](#_bookmark44) Combined with the big news happened related with coronavirus and KDE result, the answers to research question 2 can be conducted:

On day 20 (January 20), the novel coronavirus pneumonia was introduced into class B infectious diseases and was managed according to class A infectious diseases (National Health Commission of the People’s Republic of China, 2020).

On day 23 (January 23), the central government of China imposed a lockdown in Wuhan in an effort to quarantine the center of an outbreak of coronavirus (Reuters, 2020).

On day 38 (February 7), doctor Wenliang Li, who was a Chinese [ophthalmologist](https://en.wikipedia.org/wiki/Ophthalmologist) and warned his colleagues that an illness is similar to Severe Acute Respiratory Syndrome (SARS) may break out, died of COVID-19 (Sina Corp, 2020). On January 3, Wuhan police summoned and admonished him for "making false comments on the Internet" (Jianxing, 2020). Li contracted the virus from an infected patient in the hospital he worked. Li died of the disease on February 7, 2020 because of coronavirus, at age 33 (BBC News, 2020). The rate of negative emotion reached its peak because of Li’s death.

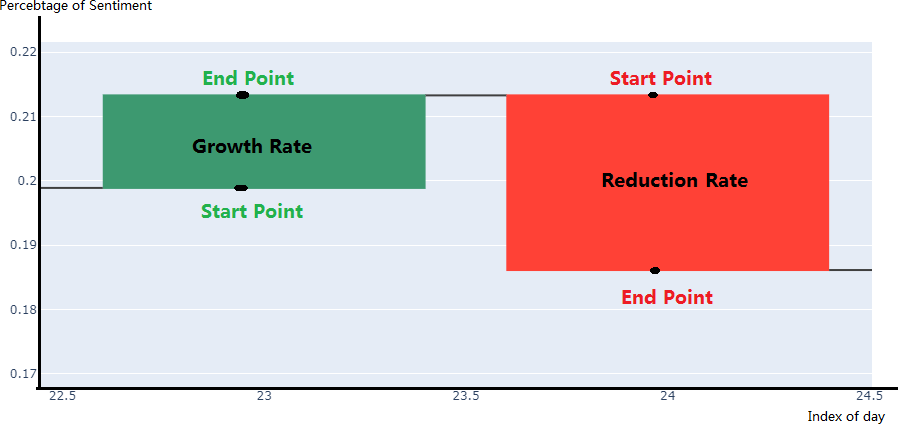


* + 1. Labeled data. (b) All data.

**Figure 13**. KDE estimates of sentiments.

## Waterfall Chart

In training the sentiment classification model on labeled data, unlabeled data are assigned a sentiment label by sentiment classification model. So there are a total of 1 million microblog posts with a label in the end.

To highlight the variation of public sentiment, a waterfall chart is used to present how each sentiment varied during this period. A waterfall chart is a special kind of column chart. Normally, we can use it to demonstrate how the starting position either increases or decreases through a series of changes.

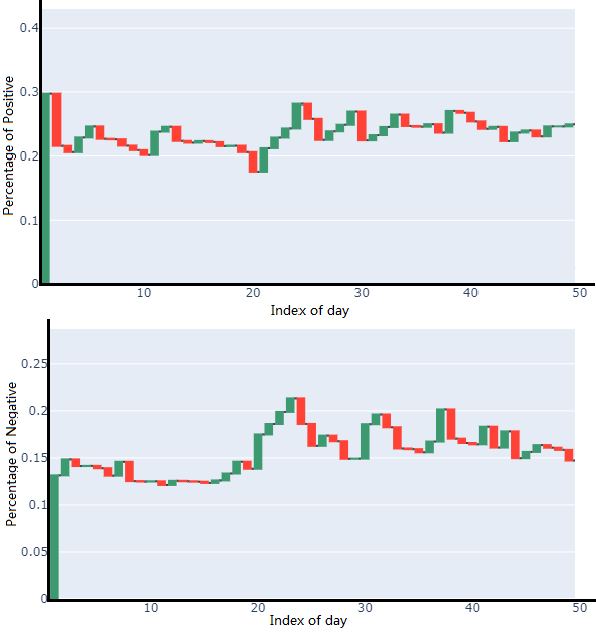
**Figure 14.** Waterfall chart with mark.

The immediate columns appear to float, and show positive and negative change from one time point to another, ending up in the final total value. Otherwise, the columns of positive and negative values need to use different color-code so that convenient for distinguishing.

We draw a chart with mark aims to help to understand the waterfall chart in [Figure 14](#_bookmark46). There are two columns: red means increasing while green means decreasing. The endpoint will be the start point of the next day. The height of the column represents the range of increase or decrease compared with yesterday. The x-axis is the index of the day and y-axis is the percentage of sentiment on this day.



**Figure 1****5**. The waterfall chart for the percentage of sentiment variation in each day based on labeled data.



**Figure 16**. The waterfall chart for the percentage of sentiment variation in each day based on all data.

## Sentiment Waterfall Charts

[Figure 15](#_bookmark47) shows the waterfall chart for sentiment variation from January 1 to February 20, which is a good answer to research question 1.

Compared with the figure of negative and positive, the percentage of people who felt neutral hold a more stable situation. For the first 11 days, the number of negative sentiments dropped significantly, from roughly 12% to 7%, then it rose suddenly. Positive sentiment is obvious grown from 20th to 24th day. After that, the number fluctuated in a large range. [Figure 16](#_bookmark48) shows how each day's percentage of sentiment varied between January 1 to February 19 based on all data (labeled data and unlabeled data). Although the percentage of positive sentiment between day 20 to day 24 performed growth, the overall figure is declined in the first 20 days. In the same period, people who felt negative are kept stable while neutral rose. In other words, some positive people true their attitude to neutral. In the next 4 days, the rate of neutral declined significantly while positive and negative increased at the same time.

After that, neutral rate fluctuated in a small range for the rest of the time. Until 30th days, the rate of negative emotion dropped to 15% roughly, while positive dropped at first but increased later. Totally, in the last 20 days, the range of varied has no significant difference. Compared to the sentiment variation of labeled data and all data, they have a different variation during those 50 days. From the waterfall charts present above, the charts based on all data show a smoother and more regular variation but some similar patterns are easier to find.

From January 18 to 24, the waterfall charts based on labeled data and all data both show that the rate of blogs with negative emotions increased suddenly.

## Discussion

Firstly, we suppose that all posts are from different users. But it is possible that someone who is very angry and published several posts to express his or her emotion. Coincidentally, those posts with the same emotion will be counted twice but only one person held negative sentiment actually.

The present work aimed at assessing how to explore the public sentiment evolution from January 1 to February 20 in China. For this purpose, BERT, which is a pre-trained language representation model was selected because of its strong text meaning extraction ability. The findings discussed demonstrate that applied BERT into sentiment classification, which might be improved in future research by more additional strategies like more training data or improved the model architecture.

With respect to pre-trained language representation models, the performance metric results support the observation that the models proposed have high performance at the text classification task. With the popularity of pre-trained language model in NLP tasks, more and more researches propose improved models based on BERT and have better performance in experiments. Thus, future research could also examine different pre-trained language representation model to extract feature from text. In our work, only 1 hidden layer with 3 units and one softmax as activation function were used as sentiment classifiers. Additionally, it is recommended that future research modifies the classifier architecture connected with pre-trained language model.

# Conclusions

With more and more people would like to share their opinion on the social platforms, which makes possible to get an insight about public sentiment using sentiment classification techniques. In this work, a sentiment classification model is built and trained on the data from microblog which is a social media platform in China, to estimate public sentiment evolution during the coronavirus outbreak period from January 1 to February 20.

The entire workflow from data to results is presented in detail. Data exploration helped to understand the dataset and provided information for the next step, data cleaning. In this step, those meaningless tokens and items with noised data will be removed. From the features of dataset, model can be designed and conducted then, used to learn sentiment classification. Finally, the sentiment classification model was applied to the processed microblog dataset to train and construct the final model. In order to evaluate the reliability of the model, different performance metrics strategies are used for performance measurement such as F1- score. When taking the results into account from the performance metric evaluation of the model, sentiment classification accuracy is approximately 75%.

Unlabeled data were annotated sentiment labels by the trained model. On the result analysis stage, the experiments were compared to the sentiment variation for labeled data and all data (labeled data and unlabeled data). Waterfall chart was choosing as a visualization strategy to present the variation of each sentiment according to its advantage for highlight variation.

For future study, it is suggested to test other pre-trained language representation models and model ensemble as more and more researches proposed improved models based on BERT which have better performance. Here, only a simple neural network with softmax is used as a sentiment classifier. It is recommended to use more complex neural networks or other technique learning models with stronger information extraction ability.

There are no more blog posts after February 20, but the trained deep learning model in this work can be used for sentiment classification for tracking the public sentiment variation on Weibo in the future.

# References

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media* (pp. 30–38). Portland, OR, USA: Association for Computational Linguistics.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1), 1-127.

Bengio, Y. (2013). Deep learning of representations: Looking forward. In *Processing of Statiscial Language and Speech Processing* (pp. 1-37). Springer Berlin Heidelberg.

Bengio, Y., Courville, A., Vincent, P., (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798– 1828. doi:10.1109/TPAMI.2013.50.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of Association for Computational Linguistics* (pp. 440-447). Prague, Czech Republic: Association for Computational Linguistics.

Che, J., Wang, J. (2014). Short-term load forecasting using a kernel-based support vector regression combination model. *Applied Energy*, 132, 602-609. doi: 0.1109/PESGM.2015.7285911.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12, 2493–2537.

BBC News (2020, February 6). Coronavirus “kills Chinese whistleblower doctor”. Retrieved 2020-02-06 from <https://aus.libguides.com/apa/apa-newspaper-web>.

Daniel, Z., Hsinchun, C., Lusch, R., Shu-Hsing, L. (2010). Social media analytics and intelligence. *IEEE Intelligent Systems*, 25, 13–16. doi: 10.1109/MIS.2010.151.

Datafountain. (2020). 疫 情 期 间 网 民 情 绪 识 别 . Retrievd 2020-03-03 from <https://www.datafountain.cn/competitions/423/datasets>.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019* (pp. 4171–4186). Minneapolis, Minnesota: Association for Computational Linguistics.

Maynard, D., & Funk, A. (2012). Automatic Detection of Political Opinions in Tweets. In *Lecture Notes in Computer Science* (pp. 88–99). Springer Berlin Heidelberg.

Diederik, P. K., Jimmy, B., (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations* . San Diego, CA, USA: Conference Track Proceedings.

Entwistle, NJ., Ramsden, P. (1983). Understanding student learning. *London: Croom Helm*.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11, 625–660.

Fan, W., Gordon, M.D. (2014). The power of social media analytics. *Communications of the ACM*, 57(6):74-81. New York, NY, USA: Association for Computing Machinery. dio: 10.1145/2602574.

Feng Z., Fu L.-W., Xiaotie D., Song H., and Lu S.-W. (2006). Automatic construction of chinese stop word list. In *Proceedings of the 5th WSEAS International Conference on Applied Computer Science* (pp. 1010–1015). Hangzhou, China: World Scientific and Engineering Academy and Society (WSEAS).

Garcaí , S., Luengo, J. and Herrera, F. (2015). Data Preprocessing in Data Mining. In

*Intelligent Systems Reference Library* (Chapter 6). Berlin: Springer.

Jianfeng G., Mu L., Andi W., and Chang-Ning H. (2005). Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4), 531– 574. dio: 10.1162/089120105775299177.

Jianxing, T. (2020). 新冠肺炎"吹哨人"李文亮：真相最重要. Caixin (in Chinese).

LeCun, Y., Bengio, Y. and Hinton, G. Deep learning. *Nature* 521**,** 436–444. dio: 10.1038/nature14539

Liu B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5, 415–463. dio: 10.1007/978-1-4614-3223-4\_13.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi- supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 17-1161). Vancouver, Canada: Association for Computational Linguistics.

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, *5*(4), 1093-1113.

Mehta, S. (2019, May 01). Developers - let's try: TensorFlow hub: A machine learning eco system. *Open Source for You,* Retrieved from [http://www.du.se.proxy.aspx/login?url=http:](http://www.du.se.proxy.aspx/login?url=http%3A//search.proquest.com/%3Furl%3Dhttps%3A//search.proquest.com/docview/2228855591%3Faccountid%3D10404)

[//search.proquest.com/?url=https://search.proquest.com/docview/2228855591?accountid=](http://www.du.se.proxy.aspx/login?url=http%3A//search.proquest.com/%3Furl%3Dhttps%3A//search.proquest.com/docview/2228855591%3Faccountid%3D10404) [10404](http://www.du.se.proxy.aspx/login?url=http%3A//search.proquest.com/%3Furl%3Dhttps%3A//search.proquest.com/docview/2228855591%3Faccountid%3D10404)

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharem agic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data,* 2(1), 1.

National Health Commission of the People’s Republic of China (2020). 2020 年 1 月全国法定传染病疫情概况. Retrieved 2020-02-29 from [http://www.nhc.gov.cn/jkj/s3578/2020](http://www.nhc.gov.cn/jkj/s3578/202002/f1dd61c00acf4e5caf2f755cc48b9063.shtml) [02/f1dd61c00acf4e5caf2f755cc48b9063.shtml](http://www.nhc.gov.cn/jkj/s3578/202002/f1dd61c00acf4e5caf2f755cc48b9063.shtml)

Oates, B. J. (2005). *Researching information systems and computing*. Sage.

Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *LREC* (pp. 1320-1326). Valletta, Malta: European Language Resources Association (ELRA).

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on empirical methods in natural language processing* (pp. 79–86). USA: Association for Computational Linguistics.

Peters, M.E., Neumann, M., Iyyer, M., et al. (2018). Deep contextualized word rep- resentations. *North American Chapter of the Association for Computational Linguistics* (pp. 2227–2237)*.* New Orleans, Louisiana: Association for Computational Linguistics.

Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL student research workshop* (pp. 43–48). Ann Arbor, Michigan: Association for Computational Linguistics.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.

Sharma, A., and Dey, S. (2012). A comparative study of feature selection and machine learning techniques for sentiment analysis. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium* (pp. 1–7). New York, NY, USA: ACM.

Sims, E. (2006). Deep Learning. *Specialist Schools and Academies Trust*.

Sina Corp, (7 February 2020). 武汉中心医院：李文亮经抢救无效去世 (in Chinese). Retrieved 2020-02-07 from [http://news.sina.com.cn/c/2020-02-07/doc-](http://news.sina.com.cn/c/2020-02-07/doc-iimxyqvz0879064.shtml) [iimxyqvz0879064.shtml.](http://news.sina.com.cn/c/2020-02-07/doc-iimxyqvz0879064.shtml)

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4), 427-437. dio: 10.1016/j.ipm.2009.03.002

Thomas, H., Christoph, W. (2007). Corruption clubs: empirical evidence from kernel density estimates. *Applied Economics*, 39, 1565-1572, dio: 10.1080/00036840500461980.

Weiguo, F., Michael, D. G. (2014). The power of social media analytics. *Communications of the ACM*. 57(6), 74-81.

Reuters, (23 January 2020)."Wuhan lockdown 'unprecedented', shows commitment to contain virus: WHO representative in China". Retrieved 2020-01-23 from <https://www.reuters.com/article/us-china-health-who-idUSKBN1ZM1G9>.

Xia, R., Zong, C. (2010). Exploring the use of word relation features for sentiment classification. In *Proceedings of the 23rd international conference on computational linguistics: Posters* (pp. 1336–1344). Stroudsburg, PA, USA: Association for Computational Linguistics.

Xipeng, Q., Tianxiang, S., Yige, X., Yunfan, S., Ning, D., and Xuanjing, H., (2020). Pre- trained Models for Natural Language Processing: A Survey. *arXiv preprint arXiv:2003.08271*.

Yuwen, Z., Zhaozhuo, X., (2018). Bert for question answering on squad 2.0. Unpublished manuscript.

Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4), 43-52. dio: 10.1007/s13042-010-0001-0.