Sentiment Analysis

Cedric Bitschene, Subir Das, Jannik Schmitt, Luana Vaduva

Outline

1. The Theory
2. The Dataset
    a. Key Points
    b. Issues with the Dataset
    c. Annotation
    d. A simple Heuristic?
3. The Approach
    a. data preparation
    b. models
        i. BERT
        ii. LSTM
4. Next Steps
5. Demo

The Theory

Sentiment Analysis in Theory

What is sentiment analysis?

- text classification task: discover opinions, classify sentiment they convey and categorize the documents
- various levels of text: document, paragraph, sentence
- aspect-based sentiment analysis: sentiments about specific characteristics of a target e.g. *battery life* or *weight* of a camera review, *plot* or *actors* of a movie review
- **polarity:** positive-negative, very positive - positive - neutral - negative - very negative
- **emotion** (objectivity/subjectivity): anger, sadness, happiness (sentiment lexicons)
- note: sentiment analysis is currently a domain-specific task. models train on one domain (e.g. movie reviews) do not transfer exceptionally well on another one (e.g. survey responses)

Sentiment Analysis in Theory

What is it used for?

- **brands**:
  - assess customers' general opinion
  - assess customers' feelings about a certain product
  - monitor reputation
  - market research
  - generally, it condenses opinions obtained from **reviews**, **surveys**, **social media** and has very broad applications
- **stock trading:** invest in companies that generate a positive sentiment
- **politics:** assess public opinion on campaign or policy announcements
- **national security:** predict and prevent terrorist attacks by analyzing dark web forums

Sentiment Analysis in Theory

Approaches

- **rule based**:
  - use sentiment words lexicon and part of speech tags
  - (adjectives, noun pairs) = object, sentiment e.g. (awful(ADJ), food (N)) = negative sentiment -> food(NEGATIVE)
  - useful for predictable texts with not much variation in grammar or semantics e.g. limited scope survey responses
  - the rules must be carefully maintained and updated and it doesn't account for the variability of language: new idioms, expressions, slang
  - implicit sentiment: "waited for an hour to be seated" = slow service (NEGATIVE)
  - ambiguous polarity words: high quality(POSITIVE) vs. high price(NEGATIVE)

Sentiment Analysis in Theory

Approaches

- **machine learning**
  - **supervised**
    - train a learning algorithm (SVM, LSTM, RNN, etc.) on a *labeled* dataset
  - **unsupervised**
    - Turney (2002)[1] classify texts as "thumbs-up" or "thumbs-down" by calculating the average semantic orientation of phrases containing adjectives and adverbs
    - Chen et al. (2019)[2] use emoticons as implicit noisy labels to learn sentiment-aware representations of text

1) Chen, Zhenpeng, et al. "SEntiMoji: an emoji-powered learning approach for sentiment analysis in software engineering." *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2019.
2) Turney, Peter D. "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews." *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002.

The Dataset

The Dataset

Key Points

- two questions, highly directional
  - answers lean strongly on the context of the questions

[Q2] Please tell us what needs to be improved.

[Q1] Please tell us what is working well.

- 17,672 answers in total
  - 8,600 answers in Q1
  - 9,075 answers in Q2

The Dataset

Issues with the Dataset

1. Some people plainly did not understand the task → nonsensical answers.

   Example - Answer to the question “What is working well?”:
   “*WORKING WELL IS DOING A GREAT JOB WITH EFFICIENCY, ACCURACY, TIMELIBESS* [sic] *AND MEETING THE EXPECTATION OF THE CUSTOMER*”

2. Many answers rely completely on the context of the question for their sentiment.

   Example - Answer to the question “What is working well?”:
   “innovation, customer relations ship [sic] and customer feedback”

   Example - Answer to the question “What needs to be improved?”:
   “Service and customer satisfaction”

The Dataset - Annotation

The Procedure

- apply sentiment label on a three-point scale
- three annotators, final label was chosen based on majority
- 300 sentences per question labelled

The Results

	Q1	Q2	total
Negative Sentiment	16	210	226
Neutral Sentiment	103	86	189
Positive Sentiment	180	4	184

Key Points:

- **60.02%** of answers in **Q1** have positive sentiment
- **70%** of answers in Q2 have negative sentiment

Easy enough for a simple heuristic?

The idea

If the questions are so highly directional, how would a classifier perform that simply assigns the “positive” label to all answers of **[Q1]** and the “negative” to all answers of **[Q2]**.

The Results *(tested on the 599 manually labelled sentences)*

Q1
True Positives = 180
False Positives = 119
Accuracy = 60.02%

Q2
True Positives = 210
False Positives = 90
Accuracy = 70%

Total
True Positives = 390
False Positives = 209
Accuracy = 65.11%

Easy enough for a simple heuristic?

The idea

If the questions are so highly directional, how would a classifier perform that simply assigns the "positive" label to all answers of **[Q1]** and the "negative" to all answers of **[Q2]**.

Keep in mind that a classification process like this is generally not useful as it **does not generalize** at all. This exercise was meant to show the biggest issue with the dataset. Furthermore, we can use these values as a benchmark for our models.

The Results (test

True Positives = 180
False Positives = 119

Accuracy = 60.02%

True Positives = 210
False Positives = 90

Accuracy = 70%

Total

True Positives = 390
False Positives = 209

Accuracy = 65.11%

The Approach

Data Preprocessing

Data preprocessing is a data mining technique that transform unstructured raw data into an understandable structured format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

Task of Data Preprocessing

- Data cleaning
- Data integration
- Data transformation
- Data reduction

Data Preprocessing Methods

Basic feature extraction

- Number of words
- Number of characters
- Number of stopwords
- Number of uppercase words
- Number of numerics

	comments	total_words	total_char	stopwords	total_uppercase	total_num
0	we do what our customers need, we communicate ...	9	60	5	0	0
1	Customs business development continues to grow...	28	161	13	0	0
2	I think the team work hard, are committed to c...	19	107	8	1	0
3	Overall working towards a customer centric env...	17	117	5	0	0
4	Customer centricity is a growing culture in th...	15	100	6	0	0

Data Preprocessing Methods

Basic Preprocessing

- Remove HTML tags
- Lower casing
- Punctuation removal
- Stopwords removal
- Remove white space
- Remove special characters
- Frequent words removal
- Rare words removal
- Tokenization
- Stemming
- Lemmatization

Tokenization, Stemming, Lemmatization

- **Tokenization :** It's a process of splitting phrase, sentence, paragraph into smaller words. Each of this word called token.

  Input : "Master in Cognitive Science"

  Output : ["Master", "in", "Cognitive", "Science"]

- **Stemming :** It's a process of removing the suffix from a word and reduce it to its root word.

  Input : "waiting"

  Output : wait

- **Lemmatization :** It's the process of grouping together the different inflected forms of a word so they can be analysed as a single item.

  Input : "Better"

  Output : Good

Tokenization, Stemming, Lemmatization

	comments	tokenization	stemming	lemmatization
0	we do what our customers need, we communicate ...	[we, do, what, our, customers, need, ,, we, co...	[we, do, what, our, custom, need, ,, we, commu...	[we, do, what, our, customer, need,, we, commu...
1	Customs business development continues to grow...	[customs, business, development, continues, to...	[custom, busi, develop, continu, to, grow, and...	[Customs, business, development, continues, to...
2	I think the team work hard, are committed to c...	[i, think, the, team, work, hard, ,, are, comm...	[I, think, the, team, work, hard, ,, are, comm...	[I, think, the, team, work, hard,, are, commit...
3	Overall working towards a customer centric env...	[overall, working, towards, a, customer, centr...	[overal, work, toward, a, custom, centric, env...	[Overall, working, towards, a, customer, centr...
4	Customer centricity is a growing culture in th...	[customer, centricity, is, a, growing, culture...	[custom, centric, is, a, grow, cultur, in, the...	[Customer, centricity, is, a, growing, culture...

Model:
BERT (Bidirectional Encoder Representations from Transformers)

Motivation

- small size of training data → transfer learning as a way to mitigate this problem
- BERT has been applied to sentiment analysis tasks before, outperforming other models on the SST-2 and SST-5 datasets[1]

Methodology

- input: semi-raw answers (spell-corrected, punctuation-corrected)

Two approaches:

1. train two models:
   a. one on answers for question a
   b. one on answers for question b
2. train one model on all answers

1: Munikar, M., Shakya, S., Shresha, A. (2019). Fine-grained Sentiment Classification using BERT. CoRR, abs/1910.03474,2019

Model:
BERT (Bidirectional Encoder Representations from Transformers)

Results (compared to simple heuristic performance)

These results were achieved by training the individual models based on the uncased base BERT model for the English language (110 million parameters).

model_q1 accuracy = 0.64 (benchmark: 0.60)
model_q2 accuracy = 0.74 (benchmark: 0.70)
average accuracy = 0.69 (benchmark: 0.65)

model_q1&2 accuracy = 0.62 (benchmark: 0.65)

Model:
BERT (Bidirectional Encoder Representations from Transformers)

Label Distribution - General Model

Negative = 0.3674
Neutral = 0.3245
Positive = 0.3081

Label Distribution - Q1 Model

Negative = 0.0000
Neutral = 0.3363
Positive = 0.6637

Label Distribution - Q2 Model

Negative = 0.7648
Neutral = 0.2352
Positive = 0.0000

Label Distribution - Complete manually labelled set

Negative = 0.3773
Neutral = 0.3155
Positive = 0.3072

Label Distribution - Manually labelled Q1

Negative = 0.0535
Neutral = 0.3445
Positive = 0.6020

Label Distribution - Manually labelled Q2

Negative = 0.7000
Neutral = 0.2867
Positive = 0.0133

Model:
BERT (Bidirectional Encoder Representations from Transformers)

Label Distribution - General Model

Negative = 0.3674
Neutral = 0.3245
Positive = 0.3081

Label Distribution - Q1 Model

Negative = 0.0000
Neutral = 0.3363
Positive = 0.6637

Label Distribution - Q2 Model

Negative = 0.7648
Neutral = 0.2352
Positive = 0.0000

Label Distribution - Complete manually labelled set

Negative = 0.3773
Neutral = 0.3155
Positive = 0.3072

Label Distribution - Manually labelled Q1

Negative = 0.0535
Neutral = 0.3445
Positive = 0.6020

Label Distribution - Manually labelled Q2

Negative = 0.7000
Neutral = 0.2867
Positive = 0.0133

Model:
BERT (Bidirectional Encoder Representations from Transformers)

Label Distribution - General Model

Negative = 0.3674
Neutral = 0.3245
Positive = 0.3081

Label Distribution - Q1 Model

Negative = 0.0000
Neutral = 0.3363
Positive = 0.6637

Label Distribution - Q2 Model

Negative = 0.7648
Neutral = 0.2352
Positive = 0.0000

Label Distribution - Complete manually labelled set

Negative = 0.3773
Neutral = 0.3155
Positive = 0.3072

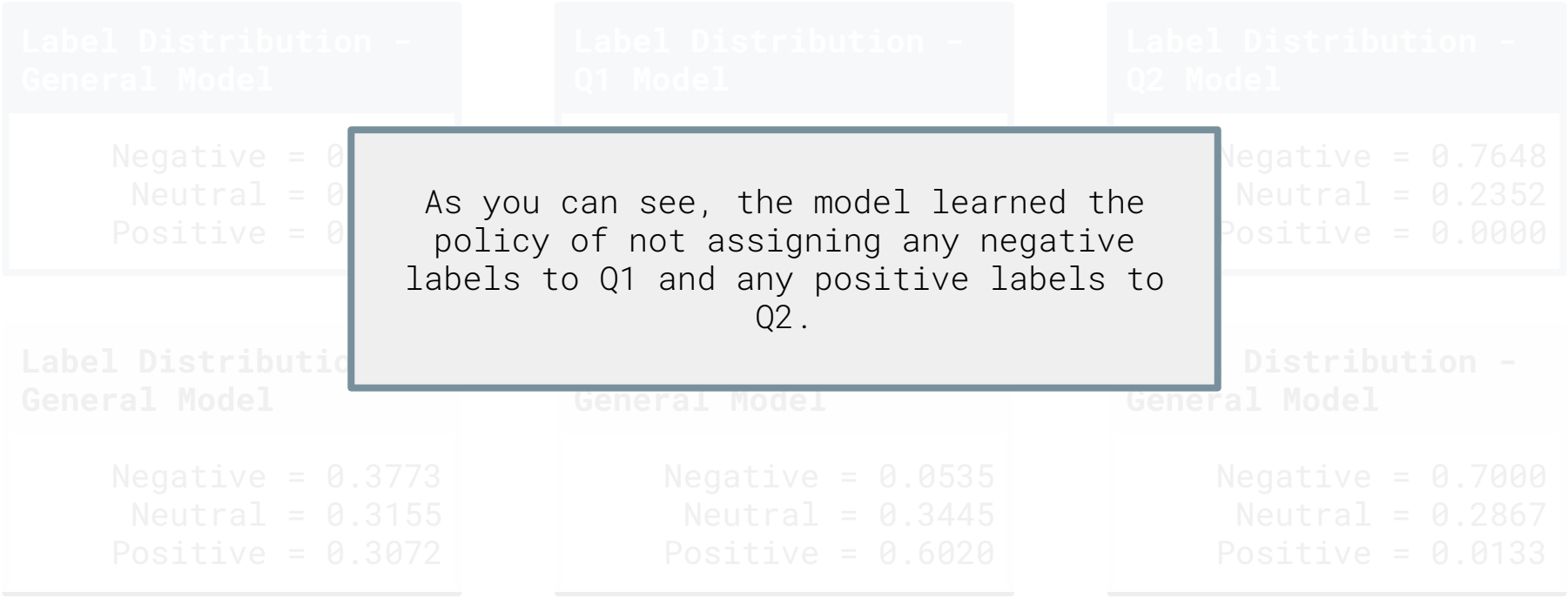
Label Distribution - Manually labelled Q1

Negative = 0.0535
Neutral = 0.3445
Positive = 0.6020

Label Distribution - Manually labelled Q2

Negative = 0.7000
Neutral = 0.2867
Positive = 0.0133

Model:
BERT (Bidirectional Encoder Representations from Transformers)



As you can see, the model learned the policy of not assigning any negative labels to Q1 and any positive labels to Q2.

Model:
BERT (Bidirectional Encoder Representations from Transformers)

Discussion

- a single model to accurately classify sentiments on the entire dataset we have remains elusive
- the performance of individual models was not good enough to warrant the exuberant amount of time and computational power that comes with using the BERT architecture

How about Alternative BERT models?

- as we know, there are newer derivations of the original BERT architecture that promise to be either:
  - more lightweight while retaining most of BERTs language understanding capabilities
  - more capable with relatively minor increases in computational demand

Model:
BERT (Bidirectional Encoder Representations from Transformers)

Unfortunately, the performance of the base model was not good enough to sacrifice some of it for computational time. Training and classification demand excessive amounts of time and resources already, meaning that we cannot afford to improve performance at the cost of computation time either. **This rules out BERT as a solution to our problem.**

Model:
LSTM (Long Short-Term Memory)

What is an LSTM

- LSTM is a recurrent neural network architecture with a memory cell that can store information for longer periods of time
- Used especially on sequential data

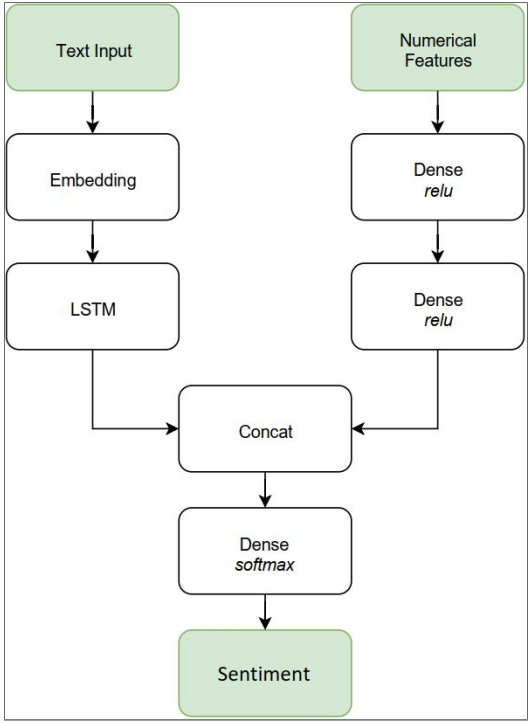
Why we chose it

- Written text is sequential data
- Consider the sentence: ‘I grew up in Germany, so I speak fluent ...’
- Sentiment is often expressed by phrases rather than single words (Wang, Xin et al., 2015)[3]
- Scalability

3) Wang, Xin, et al. "Predicting polarities of tweets by composing word embeddings with long short-term memory." *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2015.

Model Architecture

- Model is not a pure LSTM,since LSTMs work best with sequential data
- In order to use non-sequential features (punctuation, spelling-based features), we adopt a hybrid approach
- Left side input: Sequential information
- Right side input: Numerical features



Word Embeddings

We have two ways of obtaining word embeddings:

1. Twitter US Airline Sentiment

- from Crowdflower's Data for Everyone library
- contains about 15000 tweets about problems of each major US airline
- manually labeled by contributors: negative, neutral, positive

2. Pretrained Embeddings

- Word Embeddings trained using Word2Vec
- Trained on 590 million English Tweets

Model:
LSTM

Results

model_q1 accuracy = 0.64 (Bert: 0.64)
model_q2 accuracy = 0.78 (Bert: 0.74)
average accuracy = 0.71 (Bert: 0.69)

Challenges

Challenges

- the highly context-dependent answers made the task problematic
  - extract opinions related to specific predefined (or pre-extracted) aspects (e.g. “management”,”teamwork”, etc.)
- the labeled training dataset was insufficient and uncertain (the annotators chose the same labels in 73% of cases, with an Cohen’s kappa score of 0.34
- sentiment analysis is domain specific so for example word embeddings trained using twitter might not be ideal

Demo

The Next Steps