“What’s wrong with this product”- Detection of product safety issues based on information customers share online

Group 18

Supervisors: Dr. Jerry Spanakis & Dr. Caroline Cauffman

Maastricht University

Index

- Problem Introduction
- Research Questions
- Methodology
- Experiments and Evaluation
- Conclusion

“What’s wrong with this product”

Detection of product safety issues based on information customers share online

Problem Introduction

Figure 1:
https://www.shutterstock.com/image-photo/summer-beach-holiday-online-shopping-concept-461355724

- Importing from foreign markets is risky because of the lack of good control at borders (often just sampling)

- The harm caused by these products can range from small defects to choking hazards or even more danger

- E-Commerce has been growing rapidly

Context & Motivation

- Online accessible data and NLP might give more insights and is much more efficient

- Focus on online reviews for electronic goods

- Goal: detecting safety alerts using deep learning

Research questions

Research Questions

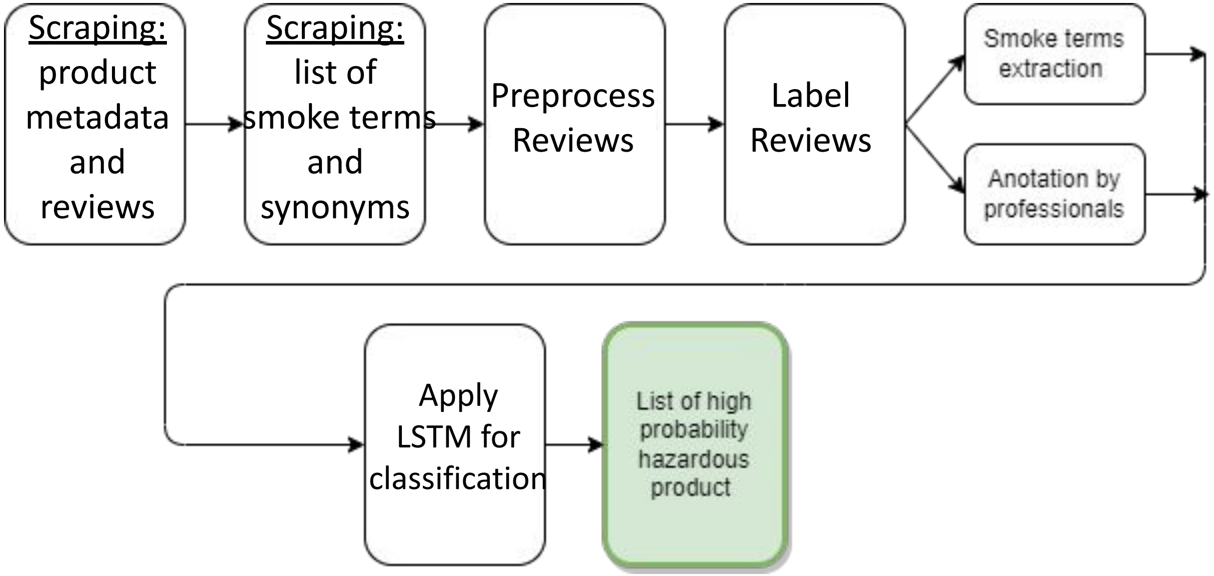
- What is necessary to obtain and prepare online data such it can be utilized for hazard detection?

- How good will perform a bidirectional LSTM in the domain of hazard detection?

- Which difficulties are arising by using this approach and from the domain?

Methodology

Overview


Scraping: product metadata and reviews
Scraping: list of smoke terms and synonyms
Preprocess Reviews
Label Reviews
Smoke terms extraction
Anotation by professionals
Apply LSTM for classification
List of high probability hazardous product

Scraping

- Many pre-existing solutions to scrape reviews, see:
  - https://github.com/search?q=amazon+scraper&type=repositories
  - https://github.com/search?q=aliexpress+scraper&type=repositorie

- Most solutions above are not sufficient because often no product metadata is given

- Solution:
  More extensive scraping tool for Amazon was implemented which can scrape additional metadata

Scraping

- Scraper applicable to specific product categories on Amazon (i.e. Sports & Outdoor instead Electronics),
  given a number of pages per category and
  number of reviews per product

- Returns a larger datatable with columns:
  product category, subcategory1, subcategory2, collected reviews, manufacturer, manufacturer country, ASIN (an unique identifier number), product url, store name, store url, etc.
  => Facilitate to trace back products

Dataset preparation

- Pre-existing large reviews database used: https://nijianmo.github.io/amazon/index.html

- Only 1-star reviews have been considered (2.7 million)

- Approach for dataset labeling:
  1. Filtering by smoke-term combinations (10760 labeled reviews)
  2. Annotation by human annotators (1200 labeled reviews)

Dataset preparation

- 2. Filtering by smoke-term combinations:

```
'smell' and 'burn' (3076)
'cable' and 'melt' (386)
'battery' and 'explod' (383)
'arcing' (97)
'electrical shock' or 'electric shock' (79)
'extremely hot' or 'extreme hot' (1260)
'extremely dangerous' (38)
'storch mark' (28)
'safety alert' (6)
```

- Extract 10 words context around terms
- Total: 5380 positive (+ 5380 negative through random sampling)

Dataset preparation

- Pre-processing steps include:
  - removal of multiple white spaces
  - removal of remaining html and url code
  - removal of punctuation
  - lowercasing
  - stopword removal (I, you, are, and, or, what, …)
  - lemmatization (i.e. drive, drove, drives become drive)
  - tokenization (split sentences and represent words by numbers)

Dataset preparation

- Example pre-processed reviews:
  "Gets extremely hot to the point that it would burn my fingers."
  => get extremely hot point would burn fingers
- "Poor quality. Only lasted for 8 months."
  => poor quality lasted for 8 months

```
negative;buy item broke onetime use would rather spend money buy something last
positive;stars yesterdaynnhowever today started selling burning smell throughout living room discovered coming
negative;purchased nike sportwatch got home plugged followed necessary instructionsand watch shut instantly
positive;immediately became hot gave burning electrical smell dangerous made china radio went
positive;looked like exploded bent screen cracked turn either battery dead gone hard
```

Model

- Applying bidirectional LSTM for context-based text classification
- => Input: sequences are truncated and padded to 12 tokens
- Model architecture:
  - Embedding layer (tokens are transformed to vectors)
  - Bidirectional LSTM layer
  - Dense layer
- => Output: prediction returns true (contains safety alert) or false

Experiments

Experiments

```
1:
Input: It already became extremely hot several times!
Preprocessed: already become extremely hot several time
Output: 0.97786075 -> positive (correct)

2:
Input: I can recommend this to you.
Preprocessed: recommend
Output: 0.24370304 -> negative (correct)

3:
Input: The plastic melted
Preprocessed: plastic melt
Output: 0.76662225 -> positive (correct)

4:
Input: The covering melted
Preprocessed: covering melt
Output: 0.18281578 -> negative (incorrect)
```

Experiments

```
5:
Input: I smelled fire coming out from my kitchen.
       As it turns out this thing caused it.
Preprocessed: smell fire come kitchen turn thing cause
Output: 0.99969393 -> positive (correct)

6:
Input: I saw fire coming out from my kitchen.
       As it turns out this thing caused it.
Preprocessed: see fire come kitchen turn thing cause
Output: 0.00180768 -> negative (incorrect)

7:
Input: Warning!! Do not purchase this. I cut my finger with this.
Preprocessed: warning purchase cut finger
Output: 0.27615544 -> negative (incorrect)

8:
Input: The cable melted in the case. I can not recommend to buy this.
Preprocessed: cable melt case recommend buy
Output: 0.02777535 -> negative (incorrect)
```

Experiments

```
9:
Input: The cable is extremely robust. I can recommend to buy this.
Preprocessed: cable extremely robust recommend buy
Output: 0.8398432 -> positive (incorrect)

10:
Input: Extremely impressed
Preprocessed: extremely impress
Output: 0.730348 -> positive (incorrect)

11:
Input: Each time I plug this I get an electrical shock!
Preprocessed: time plug get electrical shock
Output: 0.16358434 -> negative (incorrect)

12:
Input: The battery exploded!
Preprocessed: battery explode
Output: 0.86267513 -> positive (correct)
```

Model Evaluation

- It mainly learns the filtered words, i.e. see 9 & 10 ‘extremely’
- Still can not recognize ‘electrical shock’ in 11 -> too few samples (79)

Reminder:

’smell’ and ’burn’ (3076)
’cable’ and ’melt’ (386)
’battery’ and ’explod’ (383)
’arcing’ (97)
’electrical shock’ or ’electric shock’ (79)
’extremely hot’ or ’extreme hot’ (1260)
’extremely dangerous’ (38)
’storch mark’ (28)
’safety alert’ (6)

- Although some context it learned, i.e. see 3 & 4 ‘plastic’

Conclusion

- More training data is needed

- Training data need to be prepared differently
  (i.e. not truncating to 12 words of context for filtered reviews)

```
positive;reception average best fact ran extremely hot decided return well glad ended
negative;got dad phone seems happy think okay would purchase
positive;smell plastic burning useless ok 2 months suddenly
negative;purchased simrad go7 sxe october 2017 added navionics platinum plus card complete
positive;thig junk computer recognize get extremely hot
```

- Paraphrasing words / sentences

Conclusion

- Train word embeddings on bigger review corpus
- Training should also consider positive reviews
- Model does only work for the electronic domain,
  for other categories other smoke terms
  (i.e. chemical, drowing) for filtering process
  or more annotated data are needed

Thank you. :)

Questions?