Aspect Term Extraction Using the spaCy NLP Library

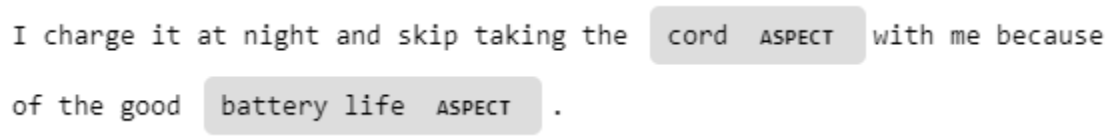
Alex Heaton
aheaton@berkeley.edu

Abstract: This paper describes techniques to use the spaCy NLP library to perform keyword (aspect) extraction from user-generated online reviews. Multiple spaCy-based models (rule-based, pre-trained, and deep learning) were evaluated, and custom deep-learning models trained on a labeled dataset provided the best overall performance.

Introduction

Aspect Extraction

Aspect extraction is a common technique to summarize and analyze user-generated feedback such as online product ratings or customer feedback surveys. "Aspects" can identify what feature or characteristics of the product users are commenting on such as the "price" or "performance" of an app or the "service" of a restaurant. The chart below shows a sample from an online review of a laptop with the aspect entities highlights.



After extracting aspects, you can use them as a dimension for additional analysis. For example, to create rankings of the top issues that customers are commenting on. By using an open-ended model to extract terms, as opposed to starting from a predefined list of terms, you can find latent issues that impact customer's experience of your products or service.

Datasets Used

A prerequisite for most deep-learning and NLP projects is to identify a ground truth data set that can be used to train and evaluate models. I used 3 datasets that were published by John Pavlopoulos and Ion Androutsopoulos as part of their research, "Aspect Term Extraction for Sentiment Analysis: New Datasets, New Evaluation Measures and an Improved Unsupervised Method". The data sets contain online reviews for laptops, restaurants, and hotels that were annotated by humans to identify the aspects contained in each sentence[1].

[1] The laptop and hotels datasets are available at https://alt.qcri.org/semeval2014/task4. The hotels dataset was provided to me by the authors at http://nlp.cs.aueb.gr/software_and_datasets/hotels.xml.

An example from one of the XML files in the laptop data shows a sentence what was labeled with 2 aspects: “cord” and “battery life”.

```
<sentence id="2339">
   <text>I charge it at night and skip taking the cord with me because of the
   good battery life.</text>
     <aspectTerms>
       <aspectTerm term="cord" polarity="neutral" from="41" to="45"/>
       <aspectTerm term="battery life" polarity="positive" from="74" to="86"/>
     </aspectTerms>
</sentence>
```

Accompanying the datasets are guidelines for the human annotators that explain what terms should be extracted as aspects and which should not. Aspects are typically nouns and are not to include subjective terms such as “bad” or “slow”. Another rule is that the category name itself such as “laptop” or manufacture names such as “Apple” should not be included as aspects. In practice, the type of information that should be extracted could vary from project to project. The techniques described in this paper could work with different types of annotation rules, such as including subjective adjectives. Though, for this research, I tried to match the definitions described in the annotations guidelines so that I could optimize the model’s ability to predict the aspects identified in the sample datasets.

SpaCy NLP Library

spaCy (https://spacy.io/), is a free, open-source, Python-based library that can be used for numerous Natural Language Processing (NLP) tasks such as entity recognition, part of speech tagging, lemmatization, and more. In this project I am taking advantage of spaCy’s named entity recognition (NER) capabilities and effectively creating a custom entity type—an aspect—that is extracted from the source text. By using spaCy’s built-in NER system I can take advantage of other tools in the spaCy ecosystem, such as displaCy, which can be used to visualize named entities discovered in a text.

Methods Used

Model 1 (Baseline): Term-Based Model

The baseline is a simple model that makes a collection of the aspects identified in the training set, extracts those terms when they appear in the sentences from the test data set, and compares the extracted terms to the aspect terms in the labelled test dataset. Even though this does not require advanced NLP, spaCy contains features that can create and evaluate these types of models. This baseline model has the following average scores across the datasets: precision=54.6, recall=58.0, and f1=53.9.

Model 2: Rule-Based Noun-Extraction Model

Since most aspects are nouns, one approach to identify aspects is to simply extract all of the nouns. This can be accomplished using spaCy’s part of speech (POS) tagger. SpaCy provides a pre-trained model that for the part of speech tagging that will predict the correct POS tag from the 17 Universal Part of Speech Tags (noun, verb, etc). This is a neural network model training on many samples of text that will consider the context that the word is used in to determine if it is a noun. This model adds a second rule to remove nouns that represent the category that the reviews are about, such as “laptops”. (This is one of the requirements from the tagging guidelines.)

This rule-based model has the following average scored across the datasets: precision=31.3, recall=62.4, and 40.8. The model performs significantly worse than the term-based model on precision and f1 score, but slightly better on recall.

Model 3: Training a Model

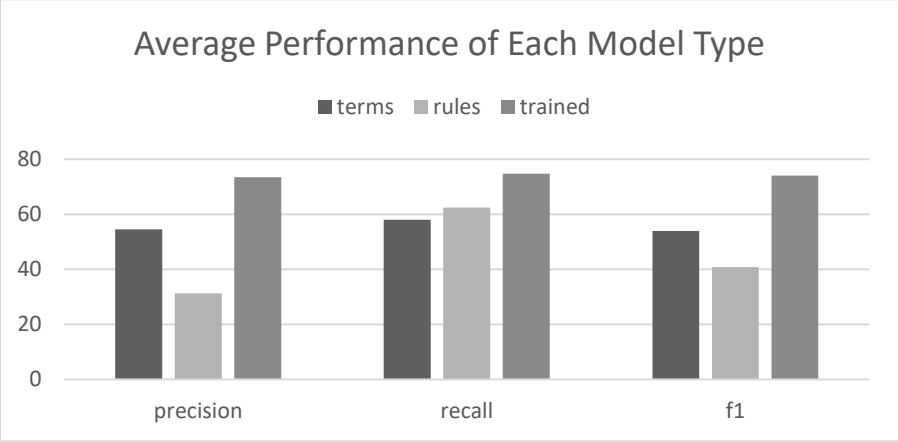
Spacy can be used to train new neural network models that identify entities in text. In this example, we are training it to extracts aspects. The training data contains the source text and a dictionary of the entities contained in that text (as identified by the human labelers). See example below:

> ("I charge it at night and skip taking the **cord** with me because of the good **battery life**.", {'entities': [(41, 45, 'ASPECT'), (74, 86, 'ASPECT')]})

It is not necessary to encode specific rules—such as to look for nouns instead of verbs—because the model will learn what an aspect should be based on the examples and their context in the training data. The trained neural network performs better than the term and rule-based models on all dimensions: precision=73.4, recall=74.7, and F1=74.

Evaluation

The table below shows how each model scored on each dataset.



The trained model performed highest on all measures. The rules-based model performs poorly on precision, presumably because it extracts all nouns including those are not relevant aspect terms. Though, that is why it performs well on recall, while the term-based model can only identify terms that were in the training data.

To understand how well each model works in practice we should look not only at the scores but the actual terms that are extracted. The table below shows the top 5 most frequent aspects from the laptops and hotels data sets—comparing the actuals in the test data set to the output of the 3 models.

hotel data	term model	rule model	trained model
food	food	food	food
service	service	service	service
atmosphere	good	place	atmosphere
staff	place	staff	staff
menu	staff	menu	menu,sushi
Matched	3/5	4/5	5/5

laptop data	term model	rule model	trained model
price	windows	price	price
performance	price	screen	performance
works	good	performance	windows
os	features	battery	works
features	quality	drive	features
Matched	2/5	2/5	4/5

Again, the trained, deep-learning models performed the best. The trained model matched the top 5 terms in the exact same order of frequency as the hotels test data set. (Menu and sushi were tied in the trained model). All models scored worse on the laptop dataset. However, they all produced logical, relevant categories. Subjectively, I prefer the categories generated by the rule-based model which identifies more specific features of the laptop such as 'screen', 'battery', and 'drive' and avoids generic terms such as 'works' and 'features'.

Building a Portable Model

The ideal model would be a generic model that could be applied to any dataset without domain specific pre-labeling and training. I will refer to these as "portable" models that can be trained on sample datasets and then used on other datasets from a different source. To test model-portability, I trained models on 2 datasets (for example hotels and restaurants) and tested them on a third (laptops). These models performed poorly when run on the data set that they were not trained on.

dataset	trained on	precision	recall	f1
restaurant	hotel+laptop	72.3	33.1	45.4
hotel	restaurant+laptop	48.5	62.1	54.5
laptops	restaurant+hotel	41.9	12.3	19.0
	average	54.3	35.8	39.6

The average f1 score was only 39.6 compared to 74 for the models that were trained on same topic as the test data. The scores was comparable to the average f1 score of the rule-based model (40.8) but the model trained on the restaurant and hotels datasets performed very poorly when applied to the laptop dataset (f1=19). The recall was especially poor. Of the top 5 aspects label in the test data set, only one "price" was extracted by the model.

Recommendations

All of the models (not including the portable model experiments) produced relevant results that would

help analysts quickly summarize the topics in user generated feedback. The rule-based model has the advantage that it does not require any pre-labeled data for training and can therefore be applied to any data set. I would recommend a using a rule-based model for an adhoc analysis or for exploratory analysis in the early phase of a project.

For an ongoing project, such as to create a system to analyze feedback on your particular product or service, I would invest in training a custom model specific to your domain to get the best results. This is not as daunting as it may sound. The creators of spaCy have also created a data annotation tool called Prodigy (https://prodi.gy/) that can be used to label samples of data that you have collected and store labels in the format used to train new spaCy models.

Acknowledgements

Thank you to John Pavlopoulos and Ion Androutsopoulos made these labelled datasets available which made this research possible. I contacted them directly and they provided me with an additional dataset, the “hotels” dataset which was not previously published.

About the Paper

This paper was created as a final project for the class Natural Language Processing with Deep Learning as part of the Master of Information and Data Science (MIDS) program from UC Berkeley. In my work as a data analyst at Microsoft I analyze customer feedback on Microsoft products and services include Microsoft Teams and the Microsoft online store. My goal was to develop a reproducible technique for aspect extraction that I can use for my own analysis and to make it available to others. Feel free to contact me if you have any questions or feedback.

References

A. John Pavlopoulos and Ion Androutsopoulos. 2014. Aspect Term Extraction for Sentiment Analysis: New Datasets, New Evaluation Measures and an Improved Unsupervised Method. In Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM) for Association for Computational Linguistics.