Information Extraction on Vehicle Postings

Chaoran Huang
University of California, Irvine
Irvine, California
chaorah1@uci.edu

***Abstract*— In this paper, I apply a combination of natural language processing and heuristic fuzzy search algorithm to a named entity recognition (NER) task to extract information from online automotive postings. This software can make it easier for purchasers to find the vehicles they want to buy, allowing them to make more informed selections.**

Keywords—NLP, NER, spaCy, visualization, transformers, vehicle

I. Introduction

A. Motivation

It was a real and overwhelming situation that author was browsing and searching for vehicles on websites like Craigslist, it is easily to face a difficulty that there are too many incomplete postings that missing key attributes: make, model and trim. Without these attributes, author need to carefully read postings to retrieve these attributes and it's also common situation that even if a make and a model is easy to know, a trim is not clearly specified which brings more difficulties on evaluating a vehicle. Meanwhile, author who is not an expert on vehicle, can hardly evaluate a vehicle's value which lead to a struggling on making decisions. In general, it will be helpful for buyers who want to buy vehicles from online postings if there is a mechanism to automatically recognize a vehicle's make, model, and trim with an estimated market price. With the aim of simplifying this process, through our NER model, buyers could facilitate evaluation of vehicles at a quick glance, thereby reducing the labor and time required/consumed in selecting candidates among enormous and overwhelming vehicles.

B. Problem Definition

This problem is a typical NER problem which is a subtask of information extraction that categorize named entities from unstructured texts which are vehicle postings into self-defined classifications. Therefore, the goal of this task is to extract four major attributes: YEAR, MAKE, MODEL, and TRIM, and two supplement attributes: VIN and ASKING PRICE. After extracting major entities, it is necessary to normalize the result to a regulated format, for example, BMW 320i should be regulated as BMW 3-series 320i with the correct trim, so that it will be more efficient and effective when evaluating vehicles' market prices. In other words, it should be useful to optimize a search engine algorithm as well.

C. Approach

The approach will be a learning-based one which requires to train convolutional neural network model to learn the patterns and probabilities of occurring entities. The system is compounded of two different modules: Entity Recognizer and Search Engine.

Entity Recognizer: this module will mainly depend on a Natural Language Processing framework: spaCy.

Search Engine: this module utilizes a combination of computing token similarities and a fuzzy search algorithm based on Levenshtein Distance.

D. Related Work

Information Extraction and NER are mechanisms that widely used in real world business. Entities are words in a text that correspond to a specific type of data. [1] Named entity recognition, or NER, is the process by which a system takes an input of unstructured data (a text) and outputs structured data, specifically the identification of entities. [1] NER systems for English are able to achieve a near-human performance. For example, the mechanism proposed MUC-7 in 1997 already scored 93.39% of F-measure which is close to the human annotators' score: 97.6% and 96.95%. [2] There are many different approaches to solve this problem. For example, Hidden Markov Chain model by Kushmerick et.al [3] proposed near 20 years ago, Conditional Random Fields model proposed for NER [4] and Transformer-based models are also worth experimenting [5]. Given to our limited computing resources and datasets. I choose spaCy which is a framework based on a convolutional neural network as my main experimental platform. Beyond of detecting entities, I furtherly provide a normalize way, a simple search engine, to map detected entity to a regulated vehicle make, model, trim for better price research and other usages.

II. Dataset

A. Structure

My dataset is collected mainly from the "cars & trucks -by owner | Craigslist" sections of Craigslist which is an advertising website. The regions I choose to target is the Southern California area. From each website' HTML, I fetched the posting title and body. These postings are published by private sellers. It turned out to be unstructured and typical.

B. Dataset statistics Analysis

I annotated vehicle postings via Doccano, an open-source text annotation tool. Due to limited labor and experience from other NER projects, for example, Foley, et. al [6] used 1608 documents in this CRF based NER model, I annotated 2295 postings which is a relatively small dataset but already enough for my task. I split this self-annotated dataset. It consists of 1836 postings for training and 459 postings for validation. To have a glance of the fundamental characteristics of the dataset, it is important to visualize them to get some insights. There are 7 classes (Fig. 1) represents the most important attributes in a posting to describe a vehicle for selling.

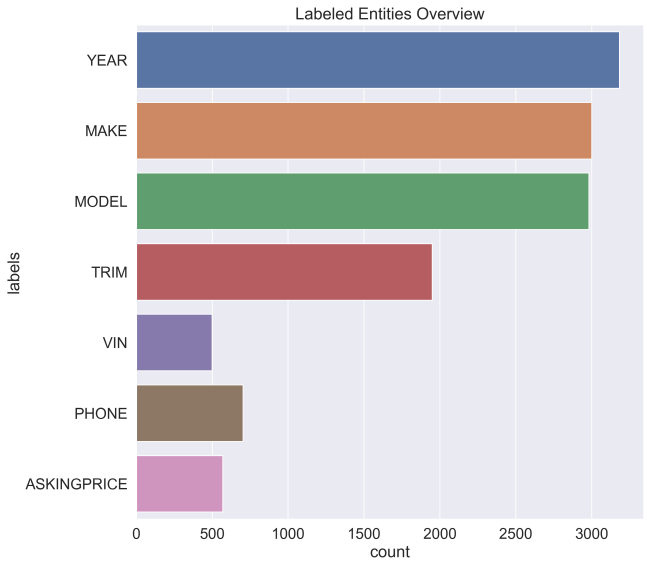

Figure 1: Dataset Entities Overview

To explore some fundamental characteristics of the posting dataset. I visualized the number of characters/words present in each posting, calculated occurrences of non-stop/stop words and explored the n-grams which defined as contiguous sequences of n words (Fig. 2).

Figure 2: Dataset Analysis

C. Challenges

Although this is a relatively simple and straightforward dataset, there are some difficulties and ambiguities exist.

In this post, there are several numbers exist. Although the asking price is specified in the title, two more possible asking price appears in the body. It may cause ambiguous when facing 99K and 4K.

1998 XK8 Convertible - Gorgeous Carnival red 99K miles super condition - $10,995 (Laguna Hills)

I am selling this gorgeous 1998 XK8 Convertible. One owner car from new, 99K fully dealer serviced mileage with records. That is approximately 4K per year! It has a clean title, Carfax and a new smog. It also has the original Jaguar books.

Figure 3: A Case of Multiple numbers

Here is another potential problem may cause. In the posting title, the vehicle's make is Chevy and model is Silverado 1500. However, in the posting body, different makes and models mentioned.

Chevy Silverado 1500 LT — 2015 64k miles - $38,500 (Newport Beach)
Very clean. 2 senior owners. Tasteful upgrades. V8 gasoline. 2 wheel drive
…
Search term: truck Silverado dodge ford Toyota Tundra 2500 f150 f250

Figure 4: A Case of Multiple Vehicles

Different languages in a posting can also be a difficulty. People may use languages other than English in their postings. Fortunately, this kind of postings is the minority.

2005 Lincoln Towncar - $5,500
2005 Lincoln Town Car
4.6L V8 motor
86,000 millas originales
La pintura y la tapa de arriba está en excelente condiciones
El interior está bien limpio
Apenas le hice un tune up completo

Figure 5: A Case of Different Languages in a posting

The complicated phone number expression will lead to a challenge. Even a human annotator has difficulty in recognizing such phone numbers in Fig. 6

2014 Dodge Dart SE - $9,000 (Mission Viejo)
2014 Dodge Dart SE with bluetooth. Brand new Khumo tires. Clean smog and all issues have been repaired. More pictures available upon request. Car will be detailed before sold. Call: 4@6@1. 93&$1-8

Figure 6: A Case of Implicit Phone Number

III. Propose Method

There are some standard libraries to implement Named Entity Recognition such as Stanford NER, spaCy, and NLTK. I choose spaCy as my tool due to its modality and efficiency.

Unlike NLTK, spaCy is more widely used for production usage. [7] It still good and simple enough for my task and experiments. According to spaCy's official documents, it supports deep learning networks, for example, a convolutional neural network (CNN) and Transformers, through a library Thinc as pipelines. [8] In the following training, I tested two different pipelines: a traditional CNN model and a Transformer model and chose different parameters carefully to achieve a better performance.

A. Entity Recognizer

During the training process of spaCy's words vector-based CNN model, there are several decisions to make, including optimizer and learning rate. The first decision to make is about the optimizer. Three optimizers provided by Thinc are stochastic gradient descent (SGD), Adaptive Moment Estimation (Adam), and Rectified Adam (RAdam), a variant of Adam. I evaluated three optimizers by model performance as well as the gradient of losses into consideration. From Fig.7, Adam and RAdam optimizers have a better performance than a traditional SGD optimizer. I chose Adam as the optimizer because it is a default optimizer in spaCy NER default config file. For the learning rate, I decided to choose the default learning rate which is 0.0005 because it leads to a decent performance. Since spaCy allows parameters to be constant value or a sequence of compound values, I also embraced a technique mentioned by Smith et. al. [9] Besides decaying the learning rate during training process, I also kept increasing the batch size during training.

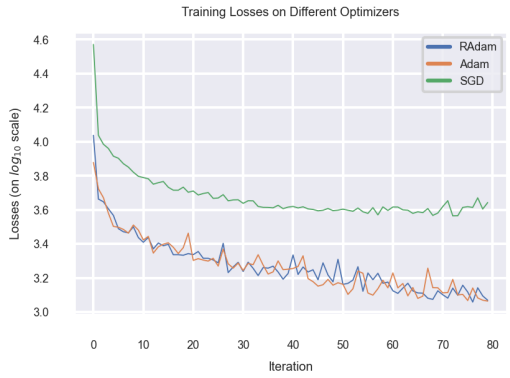

Figure 7: Training Losses on Different Optimizers

spaCy provides transformers as one of its pipes. Transformers are powerful models that can improve models' performance and it requires GPU to train and deploy. I made similar decisions on the Transformer model and performed some experiments on hyperparameters and so on.

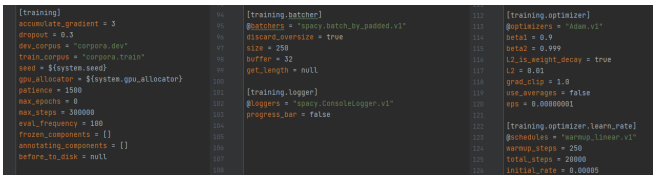

Figure 8: Transformer Model's Hyperparameters

When designing my models, some changes are also important. Tokenizer is a case deserve to be mentioned. The default tokenizer provided by spaCy takes hyphens as a delimiter. Words, for instance, Mercedes-Benz will be taken as two words: Mercedes, Benz. This is an unacceptable situation; therefore, I removed such symbols from tokenizers to achieve a more reasonable performance.

B. Search Engine

After obtained my NER of decent quality, I continued to develop a simple search engine system to map extracted major entities to a regulated vehicle type that can be easily searched online.

I obtained a json file which contains vehicles' year, make, model, trim from My Car Auction, Inc. I took this json file as my database of vehicle attributes. Systematically, after I acquired my detected vehicle's major attributes: year, make, model, trim. I narrowed down the search range by year, make, model, trim sequentially. Ideally, one or a small range of vehicle types will the result(s).

	year	make	model	trim
26794	1998	B M W	3 SERIES	328IC 2D CONVERTIBLE
20051	1994	PONTIAC	TRANS SPORT	3D MINIVAN 3.8L
48137	2007	CHEVROLET	1500 SILVERADO 2WD V8	EXT CAB 6.0L 2LT
19012	1994	FORD	EXPLORER 4WD V6	2D SUV 4.0L XLT
52831	2008	FORD	F350 DRW 2WD V10	REG CAB 6.8L XL
...	...	...	...	...
88544	2019	INFINITI	Q70L 2WD V8	4D SEDAN LUXE
3449	1984	GMC	1500 PICKUP 2WD V8	REG CAB 5.0L
45376	2006	CHEVROLET	3500 SILVERADO 2WD V8	CREW CAB 8.1L DRW LS
3753	1984	NISSAN	200 SX	2D HATCHBACK DELUXE
9787	1989	EAGLE	PREMIER	4D SEDAN LX

Figure 9: A Sample of Regulated/Formatted Vehicle Types

In addition, I take syntactical similarity and spelling difference into consideration. For the syntactical similarity, I compare detected entities to targeting vehicle attributes based on their word vectors which is a word frequency value learned across the corpus to achieve a syntactical score. Meanwhile, I applied Levenshtein distance (1) to measure the difference between two words/sentences. These two scores contribute to the confidence score of my search engine result for later validation.

$$\mathrm{lev}(a,b) = \begin{cases} |a| & \text{if } |b| = 0, \\ |b| & \text{if } |a| = 0, \\ \mathrm{lev}\left(\mathrm{tail}(a), \mathrm{tail}(b)\right) & \text{if } a[0] = b[0] \\ 1 + \min \begin{cases} \mathrm{lev}\left(\mathrm{tail}(a), b\right) \\ \mathrm{lev}\left(a, \mathrm{tail}(b)\right) \\ \mathrm{lev}\left(\mathrm{tail}(a), \mathrm{tail}(b)\right) \end{cases} & \text{otherwise,} \end{cases} \quad (1)$$

When combining these two metrics together to improve the fuzzy search engine, the result is taken syntactical and spelling into account.

Furtherly, I also validated extracted supplement entities: phone number and VIN to make sure they meet the phone and VIN number formats.

IV. Experimental Result

A. Evaluation

I use the following classic metrics to evaluate my NER. For the search engine, I take a result to be acceptable when the confidence score is larger than 0.5.

$$precision = \frac{tp}{tp + fp} \quad (2)$$

$$recall = \frac{tp}{tp + fn} \quad (3)$$

$$F1 - Measure = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \quad (4)$$

$$confidence\_score = \frac{word\_vector\_similarity + levenshtein\_distance}{2} \quad (5)$$

B. Results

My results will mainly about the CNN based Word Vector NER model and the Transformer based NER model. Under the circumstance of helping buyers extract critical information from an overwhelming postings and choices, a high recall rate can facilitate buyers more. My results from Table. 1 proved that buyers could get a decent amount of valuable information from online unstructured postings.

Table 1: NER Models' Results

NER scores (%)	Precision	Recall	F-1 Measure
CNN (Epoch 21)	80.97	81.69	81.33
Transformer (Epoch 12)	73.95	79.73	76.73

For the search engine, under the circumstance of considering a detection as a successful one when the confidence score is greater than 0.5, I tested them on 6000 cases, the success rate is 86.28%. Considering the difficulty of specifying a correct trim, this is an amazing number. This performance can truly help buyers on selection and filter vehicles to narrow down their interested types.

C. Conclusion on NER models

From my experiments, Transformer models and CNN models achieved a close performance regarding the F-measure score. However, when I am looking into the performance on detailed entities, from Table. 2, there are some interesting observations. For transformer-based model, it didn't outperform CNN in most domains in terms of F1 score. The possible reasons can be there are too few epochs trained and better hyperparameters can be found. I believe that transformer-based model has the potential to have much better performance when I have better computational resources and hyperparameters. Despite of these factors, transformer-based model has a higher recall score compares to its precision score. This observation aligns the statement made by Lothritz et. al that transformer-based models perform worse in terms of precision but better in terms of recall. [5]

Table 2: Results on Entities

		CNN			Transformer		
ID	Entity	Prec	Rec	F1	Prec	Rec	F1
1	year	84.37	93.85	88.86	87.90	88.31	88.10
2	make	84.77	91.38	87.95	86.62	88.46	87.53
3	model	80.73	82.95	81.83	71.37	84.58	77.41
4	trim	71.15	54.81	61.92	72.03	97.70	82.93
5	vin	96.59	97.70	97.14	54.72	73.11	62.59
6	price	68.63	58.82	63.35	64.46	45.68	53.47
7	phone	64.29	61.02	62.61	35.56	72.03	47.62

The entities' characteristics also influence models' performance. It is straightforward that both models have better F1 score on year, make, model, and VIN. One reasonable assumption is that year, make, model have the most annotation in the dataset which is around 3000 and VIN is a formatted string that has certain length and letters. However, for trim and phone number, these two entities have far less annotation which damages the result. Furthermore, as described in the previous Dataset session, price and phone have more ambiguities among context and complicated expression which can even cause confusion for human annotators.

V. Discussion

In this task, I completed a workflow from data' collection, cleaning and annotation to models' designing, training and evaluation. I also deployed my model on cloud service and created APIs which shown in Fig.10 so that my work can be beneficial to more people.

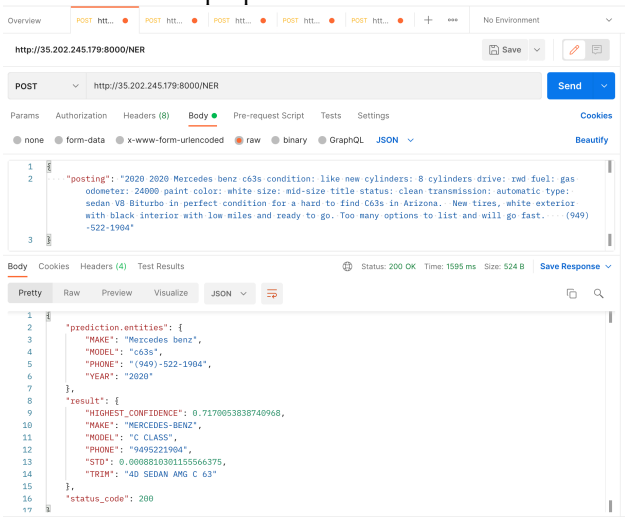

Figure 10: A Sample API Called from Postman

Nevertheless, there are some challenges of this vehicle NER.

Language habit: A typical one is that the trend of people's words/language habit continue to change. Even if the change is slight, it may affect the model performance greatly. Therefore, I should keep cultvating the dataset to attain good performance.

Ambiguity: In addition to the concern of language habit' long term shift, ambiguity is a serious issue deserves consideration. From the challenges I listed in Dataset session, there are a lot of ambiguities happen for vehicle make, model, trim and price. My solution is to enhance the quality of annotation so that models can learn better.

Linguistics: There is one final problem associated with vehicle NER and that is the problem of linguistics. As I discussed previously, it is possible to have different languages, for example, English and Spanish, in one posting. Currently, only a few words/sentences appear in the posting which is acceptable to my model for now. However, I cannot ignore the possibility that an increasing proportion of bilingual postings occur in the future. A feasible solution to handle multiple languages in a posting can be BERT and transformer-based models [10]. When a posting is half English and half Spanish, we can interpret the postings first and handle them separately.

VI. Conclusion

In this task, I presented a system that can extract vehicle attributes from online postings and format the attributes to improve vehicle value evaluation. My system achieved a decent NER success rate and is light weight to easy use which will not cause deployment problem for production.

I am excited about this software, although there are some works remain to be improved in the future, it already proved to be trustful. In the vehicle industry, AI and machine learning technology have great impact on the automobile manufactory industry like self-driving vehicles. However, in the automobile sales industry, the technology influence is relative less. I hope my system can inspire and bring some insights to others.

References

[1] W.J.B. Mattingly, "Introduction to Named Entity Recognition" Smithsonian Data Science Lab and United States Holocaust Memorial Museum, January 2021.

[2] Elaine Marsh, Dennis Perzanowski, "MUC-7 Evaluation of IE Technology: Overview of Results", 29 April 1998

[3] Kushmerik N. et al., Information Extraction by Text Classfication

[4] Sutton, Charles, and Andrew McCallum. "An Introduction to Conditional Random Fields." Foundations and trends in machine learning 14.4 (2012): 267–373.

[5] Lothritz, Cedric, Kevin Allix, Lisa Veiber, Jacques Klein, and Tegawendé François D. Assise Bissyande. "Evaluating Pretrained Transformer-based Models on the Task of Fine-Grained Named Entity Recognition." In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 3750-3760. 2020.

[6] Foley, John, Sheikh Muhammad Sarwar, and James Allan. "Named Entity Recognition with Extremely Limited Data." (2018): n. pag. Print.

[7] "Facts & Figures -spaCy" spacy.io. https://spacy.io/usage/facts-figures#other-libraries

[8] Thinc: A refreshing functional take on deep learning, compatible with your favorite libraries, Github, https://github.com/explosion/thinc

[9] Smith, Samuel L et al. "Don't Decay the Learning Rate, Increase the Batch Size." (2017): n. pag. Print.

[10] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in neural information processing systems*, pp. 5998-6008. 2017.