Ontology creation on unstructured data.

Using Natural Language Processing and Convolutional Neural Networks.

Sooraj Cheeti
DAEN
George Mason University
Fairfax, VA, USA
scheeti2@gmu.edu

Dilip Molugu
DAEN
George Mason University
Fairfax, VA, USA
dmolugu@gmu.edu

Nikhil Reddy Pathuri
DAEN
George Mason University
Fairfax, VA, USA
npathuri@gmu.edu

ABSTRACT

Usage of ontologies has increased a lot in recent years and has applications in many domains. The creation of ontologies has many applications to capture the relationships between various entities and also to increase the knowledge about a particular domain. They are also used extensively in knowledge mining. Ontologies are generally created using the tools like protege by manually writing the relationships between different entities. This process can be automated using the NLP (Natural Language Processing) techniques.

This will help us in creating more extensive ontologies from unstructured data. The difficulty in this kind of ontology generation is that it might generate some relations which might not make sense. We have tried to generalize this kind of ontology generation by observing various patterns in the data after using various techniques in NLP like lemmatization, stemming, Ngrams (unigrams, bigrams, trigrams) and POS tagging. This kind of ontology generation needs a lot EDA (Exploratory Data Analysis) to extract the important features from the raw data. The unstructured data makes it more difficult to find the relations as it contains lot of unwanted data like website links, html tags (as the data is taken from the website) and the words which simply doesn't make any sense.

As there is no proper trained data for the unstructured data, we need to create a training data from the unstructured data so that it can be used to train a neural network. Neural networks like Convolutional neural networks has been proven very useful in text classification. Therefore, we wanted to test the working of CNN on the creation of ontologies. As CNN works on the understanding the patterns in the data and gradient descent it is the best technique to find the relation in the ontology creation. The input for the neural networks and the hidden layers is given tin the form of word2vec embeddings and tfidf (term frequency–inverse document frequency). But after performing various tests using trial and error methods word2vec turned out to be a better one by giving better results.

KEYWORDS

Ontology, unstructured, data, NLP (natural language processing), Unigrams, bigrams, trigrams, POS-tagging, Word sense disambiguation, machine learning, neural networks, CNN (convolutional neural networks), tensor flow, genism, Word2vec, (tfidt) term frequency–inverse document frequency, tokenization.

1 Introduction

The Term NLP stands for Natural Language Processing/ Neuro-Linguistic Programming. It can be explained as the ability of the computer program to deal with and understand the human language and get insights from it. NLP is one among the components of Artificial Intelligence which is being widely used in today's world for the purpose of analysis is being majorly used for the purpose of Sentiment Analysis (Computational and contextual mining of text data in order to get the behavior, meaning of those particular text data towards a particular topic, business, etc.) using which data scientists can interpret data by using models and improve the business by proving insights .These techniques are majorly performed on text data such as tweets, reviews etc.

The Current approaches to NLP re based on deep learning techniques that helps to examine and retrieve patterns from the text data using labelled data to train. Use of techniques like tokenization, stemming, lemmatization and POS tagging are the very basic steps used in NLP. They are also proved to be powerful in text classification and understanding the underlying meaning in the data.

The use of Gensim has made it easy to preprocess the data such that it can be used as an input for the CNN. As the models or computer only understands the numbers and makes it possible for finding the relation between the entities. TensorFlow has been used as it uses the GPU of the computer and will make the models run faster. The processing of the models and training especially take a lot of time. For this us reason it is recommended to the use a machine with good amount of RAM and processing speed. The operations running on the model training are very complex and may take few hours sometimes depending on the number of epochs used in the model.

2 Related Work

Development of an ontology on a specific domain depends upon the perspectives of a knowledge engineer, and the output the is expected to denote a concept or element of that particular domain. So, a more systematic, efficient automated approach is possible by the automatic identification of these concepts/keywords. An idea about the most frequently used keywords within a multiline expression can help us creating a semantic network.

There are various techniques and algorithms which have been used to perform analysis on these types of text data such as:

- Latent Semantic Analysis and Singular Value Decomposition (SVD)
- Hierarchical Clustering Algorithms
- Wordnet Ontology

The latent Semantic Analysis and Singular Value Decomposition deals with set of documents. When coming to the concepts related to this particular project is about Hierarchical Clustering Algorithms and wordnet Ontology.

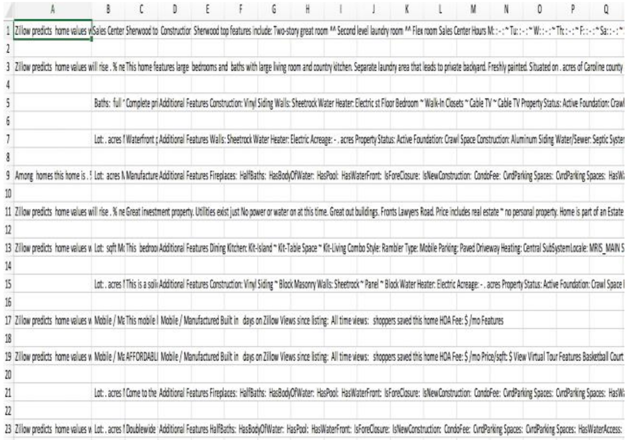
Hierarchical Clustering Algorithms: This Clustering Algorithms is used to perform an analysis on concepts obtained to generate levels of hierarchy of the ontologies. There are two ways to implement hierarchical clustering they are Bottom-up and Top down approaches. The bottom-up solution starts with individual keywords / concepts grouped together with more similar ones in order to get a single group. The top-down approach starts with all the concepts and objects in one group and are subdivided according to proximity in smaller groups.

Wordnet Ontology: Wordnet can be considered an ontology constructed in a way which can be used locally or online. It has information of nouns, adjectives, verbs and adverbs which can be used to determine semantic connections and to track the connections between morphological words. (Novelli & Parante de Oliveria, 2012)

Wordnet is used for the creation of semantic relations between the ontology concepts focusing on the creation of properties, axioms and restrictions.

3 Data preparation

The data set which is used for this project consists of text data and has various records in the form of statements written by various individuals and organizations for the purpose of business. The data set has 150 records with various aspects covered such as description of the house, location, Zillow rating, price, basic amenities, Zillow predictions, construction details, facilities available, information regarding renovation of the house if any, extra features of the property such as availability of pool etc. The data was very unstructured with 150 records and cleaning and pre-processing of the data has to be done in order to proceed further with the project.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Zillow predicts home values w	Sales Center	Sherwood to	Constructior	Sherwood top features include: Two-story great room ^^ Second level laundry room ^^ Flex room Sales Center Hours M: : - : ~ Tu: : - : ~ W: : - : ~ Th: : - : ~ F: : - : ~ Sa: : - : ~												
2																	
3	Zillow predicts home values will rise . % ne		This home features large bedrooms and baths with large living room and country kitchen. Separate laundry area that leads to private backyard. Freshly painted. Situated on . acres of Caroline county														
4																	
5		Baths: full	" Complete pri	Additional Features Construction: Vinyl Siding Walls: Sheetrock Water Heater: Electric st Floor Bedroom ~ Walk-In Closets ~ Cable TV ~ Cable TV Property Status: Active Foundation: Crawl													
6																	
7		Lot: . acres !	Waterfront ;	Additional Features Walls: Sheetrock Water Heater: Electric Acreage: - . acres Property Status: Active Foundation: Crawl Space Construction: Aluminum Siding Water/Sewer: Septic Syster													
8																	
9	Among homes this home is . !	Lot: acres N	Manufacture	Additional Features Fireplaces: HalfBaths: HasBodyOfWater: HasPool: HasWaterFront: IsForeClosure: IsNewConstruction: CondoFee: CvrdParking Spaces: CvrdParking Spaces: HasWa													
10																	
11	Zillow predicts home values will rise . % ne		Great investment property. Utilities exist just No power or water on at this time. Great out buildings. Fronts Lawyers Road. Price includes real estate ~ no personal property. Home is part of an Estate														
12																	
13	Zillow predicts home values w	Lot: sqft Mc	This bedroo	Additional Features Dining Kitchen: Kit-Island ~ Kit-Table Space ~ Kit-Living Combo Style: Rambler Type: Mobile Parking: Paved Driveway Heating: Central SubSystemLocale: MRIS_MAIN S													
14																	
15		Lot: . acres !	This is a soli	Additional Features Construction: Vinyl Siding ~ Block Masonry Walls: Sheetrock ~ Panel ~ Block Water Heater: Electric Acreage: - . acres Property Status: Active Foundation: Crawl Space I													
16																	
17	Zillow predicts home values w	Mobile / Mc	This mobile l	Mobile / Manufactured Built in days on Zillow Views since listing: All time views: shoppers saved this home HOA Fee: $ /mo Features													
18																	
19	Zillow predicts home values w	Mobile / Mc	AFFORDABLI	Mobile / Manufactured Built in days on Zillow Views since listing: All time views: shoppers saved this home HOA Fee: $ /mo Price/sqft: $ View Virtual Tour Features Basketball Court													
20																	
21		Lot: . acres !	Come to the	Additional Features Fireplaces: HalfBaths: HasBodyOfWater: HasPool: HasWaterFront: IsForeClosure: IsNewConstruction: CondoFee: CvrdParking Spaces: CvrdParking Spaces: HasWa													
22																	
23	Zillow predicts home values w	Lot: . acres !	Doublewide	Additional Features HalfBaths: HasBodyOfWater: HasPool: HasWaterFront: IsForeClosure: IsNewConstruction: CondoFee: CvrdParking Spaces: CvrdParking Spaces: HasWaterAccess:													

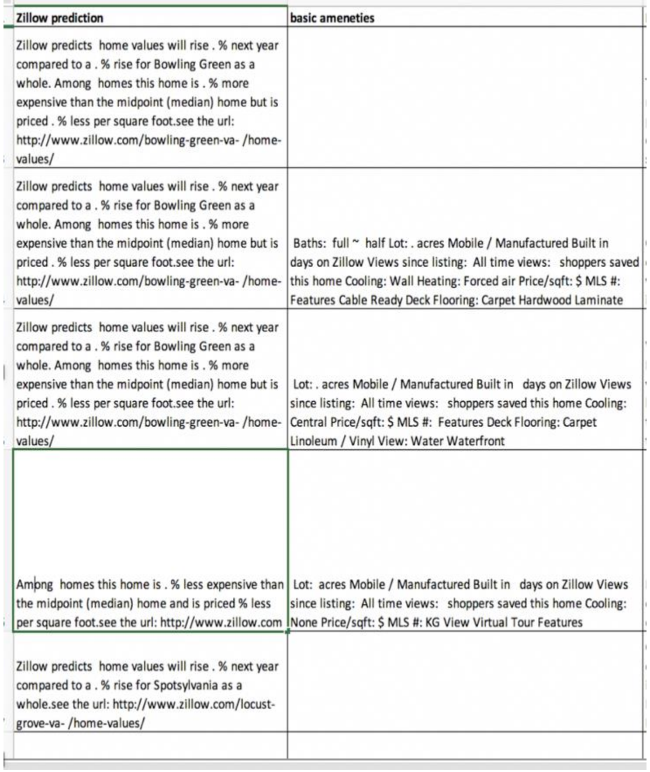
Figure 1: The snapshot of unstructured raw data.

This can be defined as the detection and removal of inappropriate, incomplete data and modifying the format of the data in order to make it more efficient by replacing the missing data and make it consistent for further use. In case of this particular data set where the data / information is in the form of text data there were a lot of things that have to be concentrated on such as some of the percentages (numeric values) for Zillow predictions was missing, empty records have to concentrated on, locations were missing, and structuring of data has to be done. We have used regular expressions in python on order to clean the data initially and structured it.

Now the cleaned data can be seen in figure 2 and figure 3 with four attributes namely Zillow Predictions, Basic Amenities, Description, Additional Features.

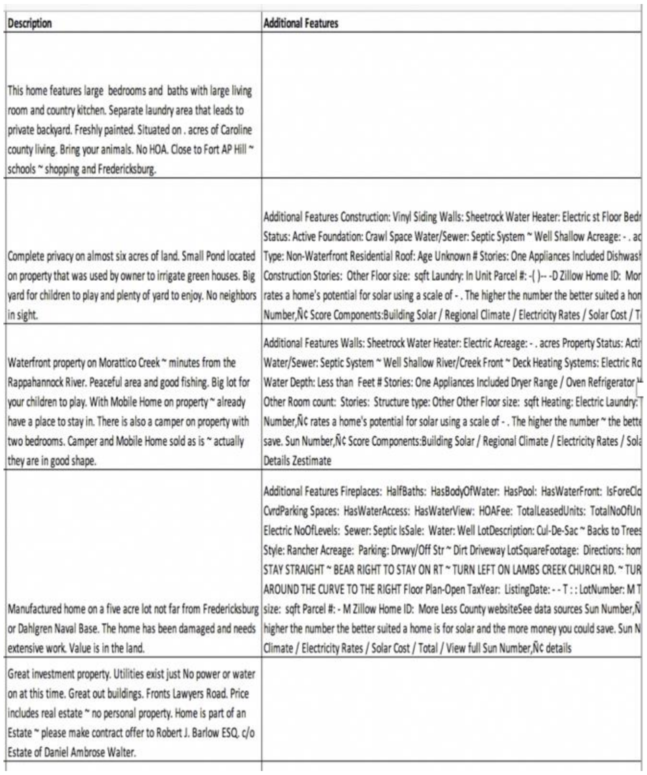
In the figure 1 we can clearly see that the data is unstructured and needs a lot of cleaning. After clear observation of the data we can clearly see that there mainly four basic columns. After going through the Zillow website and further research the data can be segregated into four different columns namely Zillow prediction, basic amenities, description, additional features. In the figure 2, the column Zillow prediction is the prediction provided by the Zillow website about the percentage rise or decrease in the price of the property in the future. the column basic amenities show the column shows all the amenities of a particular property listed on the Zillow website.

In the figure 3 we can see that description column is providing the information about the house which was provided by the owner or by the person who posted the property on the Zillow website. In the column Additional features more features like pool details, parking details, area of the property etc. are provided.



Zillow prediction	basic ameneties
Zillow predicts home values will rise . % next year compared to a . % rise for Bowling Green as a whole. Among homes this home is . % more expensive than the midpoint (median) home but is priced . % less per square foot.see the url: http://www.zillow.com/bowling-green-va- /home-values/	
Zillow predicts home values will rise . % next year compared to a . % rise for Bowling Green as a whole. Among homes this home is . % more expensive than the midpoint (median) home but is priced . % less per square foot.see the url: http://www.zillow.com/bowling-green-va- /home-values/	Baths: full ~ half Lot: . acres Mobile / Manufactured Built in days on Zillow Views since listing: All time views: shoppers saved this home Cooling: Wall Heating: Forced air Price/sqft: $ MLS #: Features Cable Ready Deck Flooring: Carpet Hardwood Laminate
Zillow predicts home values will rise . % next year compared to a . % rise for Bowling Green as a whole. Among homes this home is . % more expensive than the midpoint (median) home but is priced . % less per square foot.see the url: http://www.zillow.com/bowling-green-va- /home-values/	Lot: . acres Mobile / Manufactured Built in days on Zillow Views since listing: All time views: shoppers saved this home Cooling: Central Price/sqft: $ MLS #: Features Deck Flooring: Carpet Linoleum / Vinyl View: Water Waterfront
Among homes this home is . % less expensive than the midpoint (median) home and is priced % less per square foot.see the url: http://www.zillow.com	Lot: acres Mobile / Manufactured Built in days on Zillow Views since listing: All time views: shoppers saved this home Cooling: None Price/sqft: $ MLS #: KG View Virtual Tour Features
Zillow predicts home values will rise . % next year compared to a . % rise for Spotsylvania as a whole.see the url: http://www.zillow.com/locust-grove-va- /home-values/	

Figure 2: The snapshot of unstructured data after segregating the data according to different categories (columns) the figure shows the columns Zillow prediction and basic amenities.

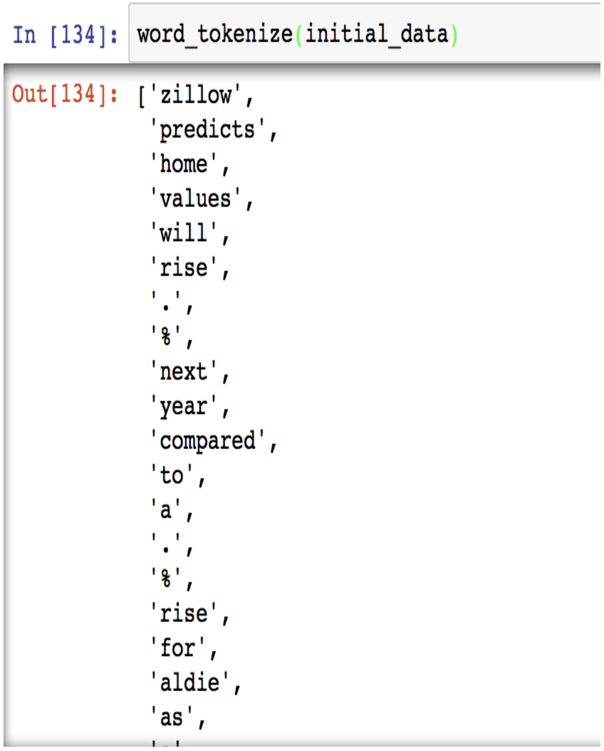


Description	Additional Features
This home features large bedrooms and baths with large living room and country kitchen. Separate laundry area that leads to private backyard. Freshly painted. Situated on . acres of Caroline county living. Bring your animals. No HOA. Close to Fort AP Hill ~ schools ~ shopping and Fredericksburg.	
Complete privacy on almost six acres of land. Small Pond located on property that was used by owner to irrigate green houses. Big yard for children to play and plenty of yard to enjoy. No neighbors in sight.	Additional Features Construction: Vinyl Siding Walls: Sheetrock Water Heater: Electric st Floor Bedr Status: Active Foundation: Crawl Space Water/Sewer: Septic System ~ Well Shallow Acreage: - . ac Type: Non-Waterfront Residential Roof: Age Unknown # Stories: One Appliances Included Dishwas Construction Stories: Other Floor size: sqft Laundry: In Unit Parcel #: -( )-- -D Zillow Home ID: Mor rates a home's potential for solar using a scale of - . The higher the number the better suited a hon Number,Ñ¢ Score Components:Building Solar / Regional Climate / Electricity Rates / Solar Cost / T
Waterfront property on Morattico Creek ~ minutes from the Rappahannock River. Peaceful area and good fishing. Big lot for your children to play. With Mobile Home on property ~ already have a place to stay in. There is also a camper on property with two bedrooms. Camper and Mobile Home sold as is ~ actually they are in good shape.	Additional Features Walls: Sheetrock Water Heater: Electric Acreage: - . acres Property Status: Acti Water/Sewer: Septic System ~ Well Shallow River/Creek Front ~ Deck Heating Systems: Electric Ro Water Depth: Less than Feet # Stories: One Appliances Included Dryer Range / Oven Refrigerator Other Room count: Stories: Structure type: Other Other Floor size: sqft Heating: Electric Laundry: T Number,Ñ¢ rates a home's potential for solar using a scale of - . The higher the number ~ the bette save. Sun Number,Ñ¢ Score Components:Building Solar / Regional Climate / Electricity Rates / Sola Details Zestimate
Manufactured home on a five acre lot not far from Fredericksburg or Dahlgren Naval Base. The home has been damaged and needs extensive work. Value is in the land.	Additional Features Fireplaces: HalfBaths: HasBodyOfWater: HasPool: HasWaterFront: IsForeClo CvrdParking Spaces: HasWaterAccess: HasWaterView: HOAFee: TotalLeasedUnits: TotalNoOfUn Electric NoOfLevels: Sewer: Septic IsSale: Water: Well LotDescription: Cul-De-Sac ~ Backs to Trees Style: Rancher Acreage: Parking: Drvwy/Off Str ~ Dirt Driveway LotSquareFootage: Directions: hom STAY STRAIGHT ~ BEAR RIGHT TO STAY ON RT ~ TURN LEFT ON LAMBS CREEK CHURCH RD. ~ TUR AROUND THE CURVE TO THE RIGHT Floor Plan-Open TaxYear: ListingDate: - - T : : LotNumber: M T size: sqft Parcel #: - M Zillow Home ID: More Less County websiteSee data sources Sun Number,Ñ higher the number the better suited a home is for solar and the more money you could save. Sun N Climate / Electricity Rates / Solar Cost / Total / View full Sun Number,Ñ¢ details
Great investment property. Utilities exist just No power or water on at this time. Great out buildings. Fronts Lawyers Road. Price includes real estate ~ no personal property. Home is part of an Estate ~ please make contract offer to Robert J. Barlow ESQ. c/o Estate of Daniel Ambrose Walter.	

Figure 3: the figure shows the columns Description and Additional features.

3.1 Cleaning of Data

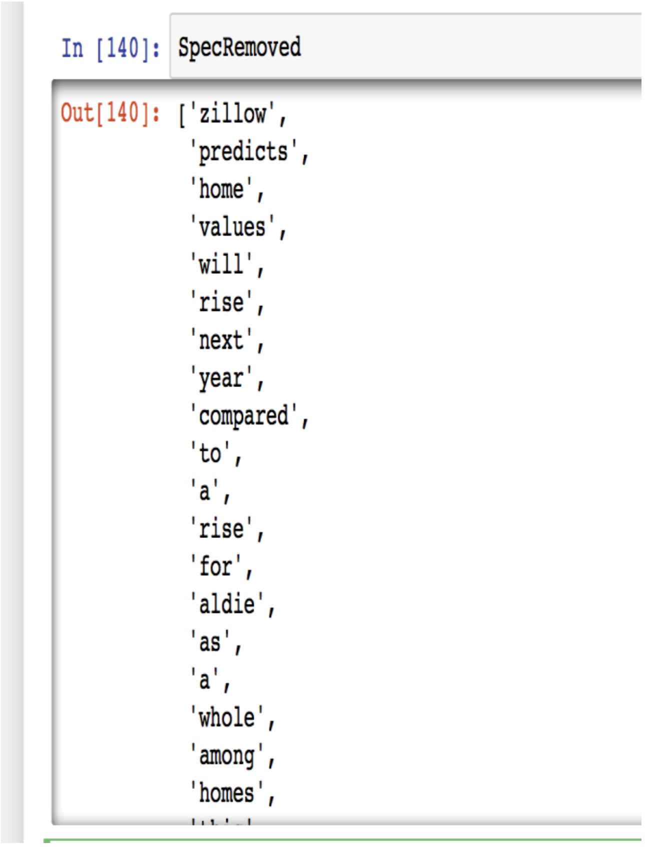
The data needs to be cleaned before we can use the NLP techniques in the data. Firstly, tokenization of the data is done. It's an act of breaking up a sequence of strings (For example a statement) into units such as words, elements, keywords called tokens. The tokenization of the data is done using the nltk package and the word_tokenize function. (Fortney, 2017)



```
In [134]: word_tokenize(initial_data)

Out[134]: ['zillow',
           'predicts',
           'home',
           'values',
           'will',
           'rise',
           '.',
           '%',
           'next',
           'year',
           'compared',
           'to',
           'a',
           '.',
           '%',
           'rise',
           'for',
           'aldie',
           'as',
```

Figure 4: the figure shows the tokenized data along with all the special characters and with stop words.

As we can see in the figure 3, after doing this process certain characters of the text data such as punctuation marks, spaces and so on were removed. So that the special characters in the data won't cause any problem in the analysis and predicting the ontologies in the data. For this we have created a list of stop words and by manually looking at the results so that they cause any problem in losing the actual sense of the data.



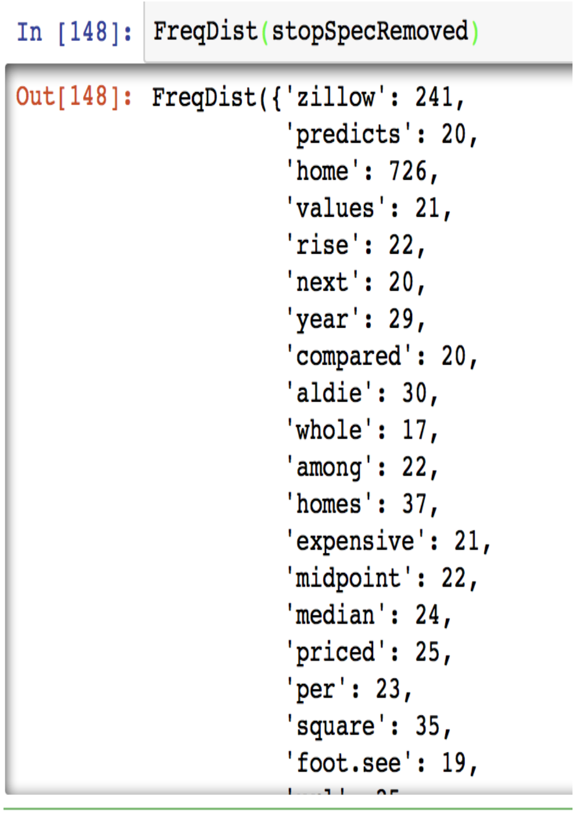
```
In [140]: SpecRemoved

Out[140]: ['zillow',
           'predicts',
           'home',
           'values',
           'will',
           'rise',
           'next',
           'year',
           'compared',
           'to',
           'a',
           'rise',
           'for',
           'aldie',
           'as',
           'a',
           'whole',
           'among',
           'homes',
```

Figure 5: the figure shows the tokenized data with the punctuations removed.

As we can see in the figure 5 the special characters we removed, and we did not use the traditional list of stop words from the nltk package. The reason for this is that removing the all the stop-words present in the nltk package will hinder us from finding the POS tags for the data and we will not be able to get the sense of the data. This is very crucial for automatic prediction of the predicates given the subject.

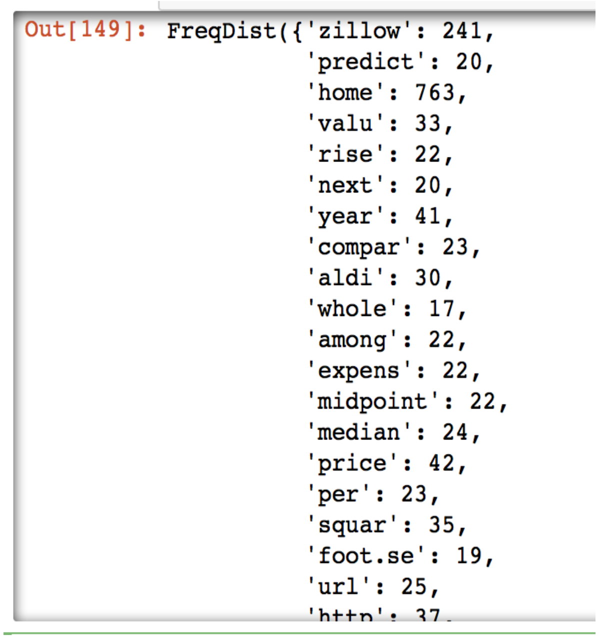
3.2 Lemmatization

Lemmatization in the domain of NLP (Natural Language Processing) refers as grouping together the different types of the words with the same root words. This is done using the NLTK package in python. Here is the example which is being shown from the figures 6 and 7.



```
In [148]: FreqDist(stopSpecRemoved)

Out[148]: FreqDist({'zillow': 241,
                    'predicts': 20,
                    'home': 726,
                    'values': 21,
                    'rise': 22,
                    'next': 20,
                    'year': 29,
                    'compared': 20,
                    'aldie': 30,
                    'whole': 17,
                    'among': 22,
                    'homes': 37,
                    'expensive': 21,
                    'midpoint': 22,
                    'median': 24,
                    'priced': 25,
                    'per': 23,
                    'square': 35,
                    'foot.see': 19,
```

Figure 6: the snapshot of the data before the lemmatization of the data.



```
Out[149]: FreqDist({'zillow': 241,
                    'predict': 20,
                    'home': 763,
                    'valu': 33,
                    'rise': 22,
                    'next': 20,
                    'year': 41,
                    'compar': 23,
                    'aldi': 30,
                    'whole': 17,
                    'among': 22,
                    'expens': 22,
                    'midpoint': 22,
                    'median': 24,
                    'price': 42,
                    'per': 23,
                    'squar': 35,
                    'foot.se': 19,
                    'url': 25,
                    'http': 37,
```

Figure 7: the snapshot of the data after the lemmatization of the data.

From the figure 6 and figure 7 we can clearly see that words like predict, home, value etc. are lemmatized. With this we can get a clear idea of important words in the data. Rather than just finding the frequency of the data which will include different forms of the same word as plurals, different tense etc.

After this step we identified the important features in the data. The figure 8 shows that the 80% of the data is covered with the top 20% of the words in the data. This is known as zipf's law. Zipf's law is usually observed in many situations and is a common thing in the real worlds. **(Piantadosi, 2015)**

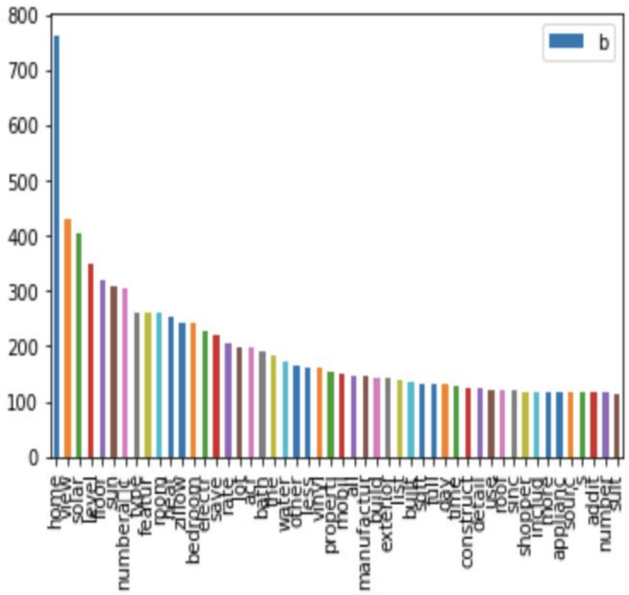


Figure 8: this plot shows that 80% of the data is covered by top 21% words.

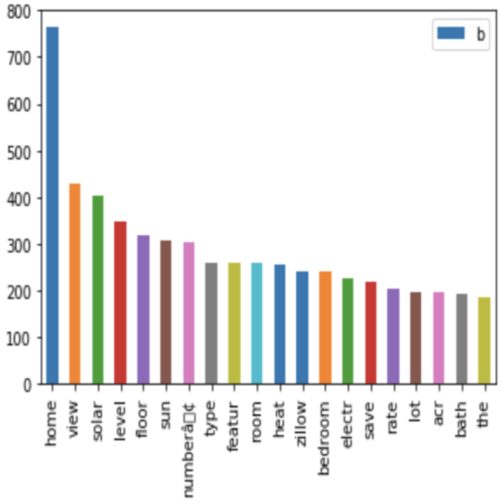
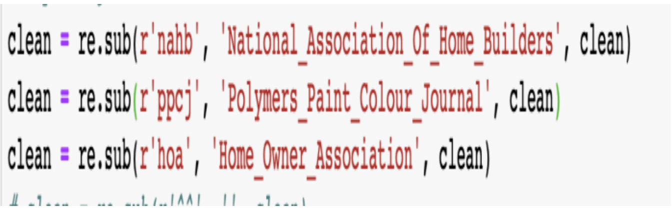


Figure 9: this plot shows the top 20 frequently used words in the data.

The figure 9 will give us the insights of the data like understanding the important words in the data so that we can understand the important features if the data. Therefore, we need to concentrate more on these features in the Exploratory analysis of our housing data.

3.3 Dealing with Acronyms

The data also has a lot of Acronyms which need to handle. We have used the regular expressions by which we able to search the data and find the acronyms in the data after we were able to find the, we have used the google knowledge base to maps those acronyms to the abbreviations. This task will give the data more meaningful information rather than using just the acronyms. For example, we have used the following mappings after we were able to identify the Acronyms. In the figure 10, some of the sample mappings are provided.



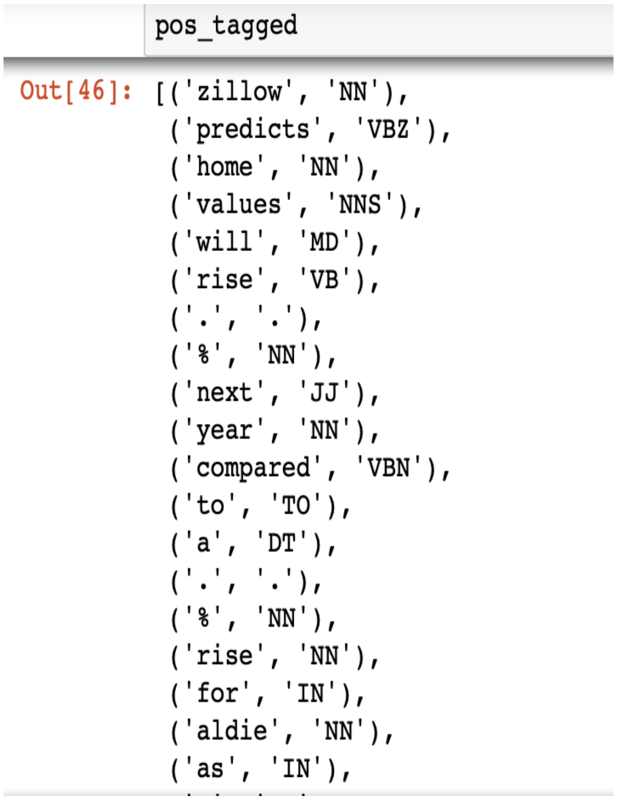
```
clean = re.sub(r'nahb', 'National_Association_Of_Home_Builders', clean)
clean = re.sub(r'ppcj', 'Polymers_Paint_Colour_Journal', clean)
clean = re.sub(r'hoa', 'Home_Owner_Association', clean)
```

Figure 10: Some of the important mappings in the data.

This kind of mappings will make ontologies more sensible as we will not be aware of all the acronyms unless someone is an expert in that particular domain. Therefore, this will allow even a normal person understand the ontologies in an effective way.

4 Preparing the training data

As there is no traditional training data for this housing data from the cleaned data. We have made a training dataset which can be used for this model. POS (parts of speech) tagging has played a major role in preparing the training data and in finding the ontologies from the data. We have tagged the data using nltk POS tagger. This has resulted in the tags for every word in the data. This can be seen in the figure 11.



```
pos_tagged

Out[46]: [('zillow', 'NN'),
          ('predicts', 'VBZ'),
          ('home', 'NN'),
          ('values', 'NNS'),
          ('will', 'MD'),
          ('rise', 'VB'),
          ('.', '.'),
          ('%', 'NN'),
          ('next', 'JJ'),
          ('year', 'NN'),
          ('compared', 'VBN'),
          ('to', 'TO'),
          ('a', 'DT'),
          ('.', '.'),
          ('%', 'NN'),
          ('rise', 'NN'),
          ('for', 'IN'),
          ('aldie', 'NN'),
          ('as', 'IN'),
```

figure 11: POS tags for all the words in the data.

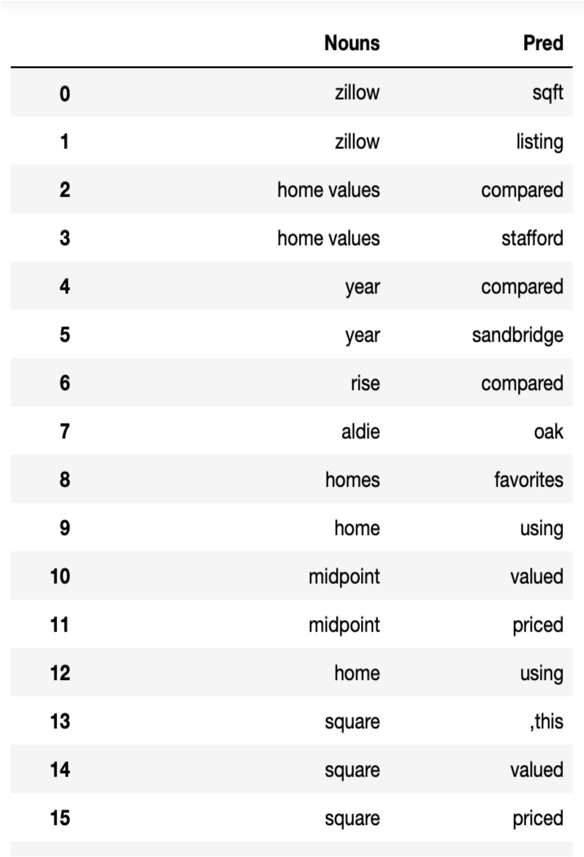
After tagging all the words, a list of all the nouns is stored in a separate list and this will be used as subjects in the creation of ontologies. From the list of all the tagged words we have observed that the words with POS tags having verbs were acting as predicated.

The words with following tags are used as predicates:

- VBD (Verb, past tense).
- VBG (Verb, gerund or present participle).
- VBN (Verb, past participle).
- VBP (Verb, non-3rd person singular present).
- VBZ (Verb, 3rd person singular present).

4.2 Creating the training data

Now, from the text all the words containing tags as nouns are extracted. The reason for this is that we are going to find the predicate from the object. In many situations we have seen that the nouns act as subject or predicate. Therefore, we have decided to use nouns as the subject. Now for every noun the predicate is found by using the word similarity. The words with tags VBD, VBG, VBN, VBP and VBZ are considered as predicates and a list is formed, which will be used as a training data for the model.



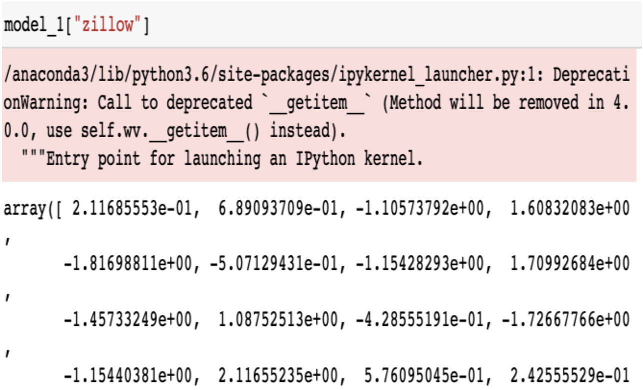
	Nouns	Pred
0	zillow	sqft
1	zillow	listing
2	home values	compared
3	home values	stafford
4	year	compared
5	year	sandbridge
6	rise	compared
7	aldie	oak
8	homes	favorites
9	home	using
10	midpoint	valued
11	midpoint	priced
12	home	using
13	square	,this
14	square	valued
15	square	priced

figure 12: Snapshot of the training data for the CNN model

In the figure we can see that for every noun the word similarity is found out. And from the list of the similar words only the most likely or similar word, which is also a verb is considered and the training data is formed.

4.3 Word2Vec embeddings

Now we have the data with the POS tags for all the words we can move forward and create a word2vec model. This will generate the word embeddings for all the words present in the data. This can also be used to get the most similar words when given a particular. This property of word2vec model is a key property in creating the ontologies. In the figure 12 we can see than given a particular word, a vector will be generated in which can be used in the training the CNN model. **(CHia, 2018)**

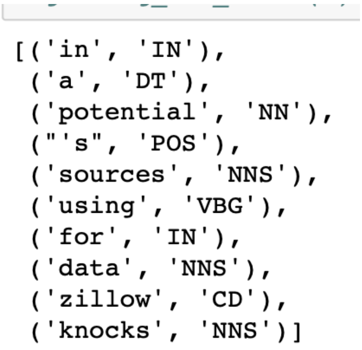


```
model_1["zillow"]

/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: Deprecati
onWarning: Call to deprecated `__getitem__` (Method will be removed in 4.
0.0, use self.wv.__getitem__() instead).
  """Entry point for launching an IPython kernel.

array([ 2.11685553e-01,  6.89093709e-01, -1.10573792e+00,  1.60832083e+00
,
       -1.81698811e+00, -5.07129431e-01, -1.15428293e+00,  1.70992684e+00
,
       -1.45733249e+00,  1.08752513e+00, -4.28555191e-01, -1.72667766e+00
,
       -1.15440381e+00,  2.11655235e+00,  5.76095045e-01,  2.42555529e-01
```

Figure 12: word vector for the word ‘Zillow’.

4.3 Using the word Similarity

The word2Vec model can be used to find the words similar to a particular word. This will help us in finding the subject, object and predicate in the creation of ontologies. In the figure 13 we can see the list of words similar to the word ‘home’.



```
[('in', 'IN'),
 ('a', 'DT'),
 ('potential', 'NN'),
 ("'s", 'POS'),
 ('sources', 'NNS'),
 ('using', 'VBG'),
 ('for', 'IN'),
 ('data', 'NNS'),
 ('zillow', 'CD'),
 ('knocks', 'NNS')]
```

Figure 13: example of word similarity for the word ‘home’.

5 CNN (Convolutional Neural Networks)

Convolution neural network is a class of neural networks which is mainly used for in analyzing visual imagery. It mainly works by learning the patterns in the images by forming multiple layered architecture. This technique was inspired from the biological processing of the humans in which the neurons in the resemble the organization of visual cortex.

CNNs in general needs less preprocessing when compared to other image classification methods. CNN in general consists of an input layer, output layer and multiple hidden layers. the hidden layers can be further divided into Convolution layers, pooling layers and fully connected layers.

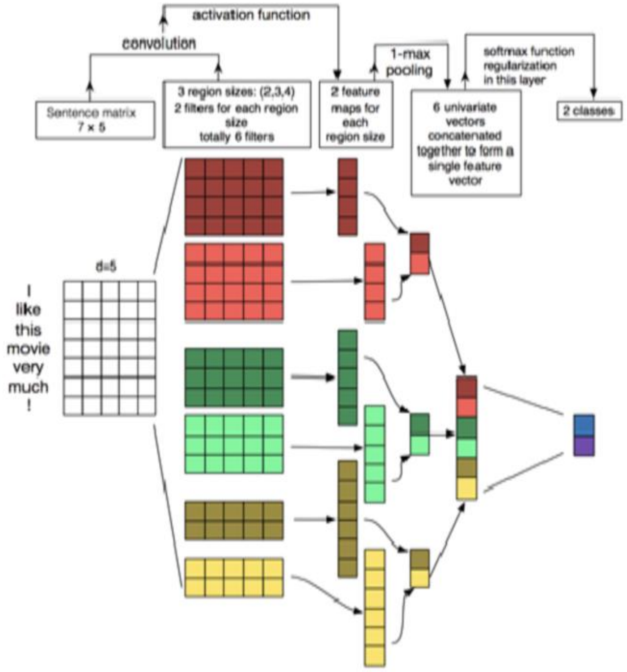
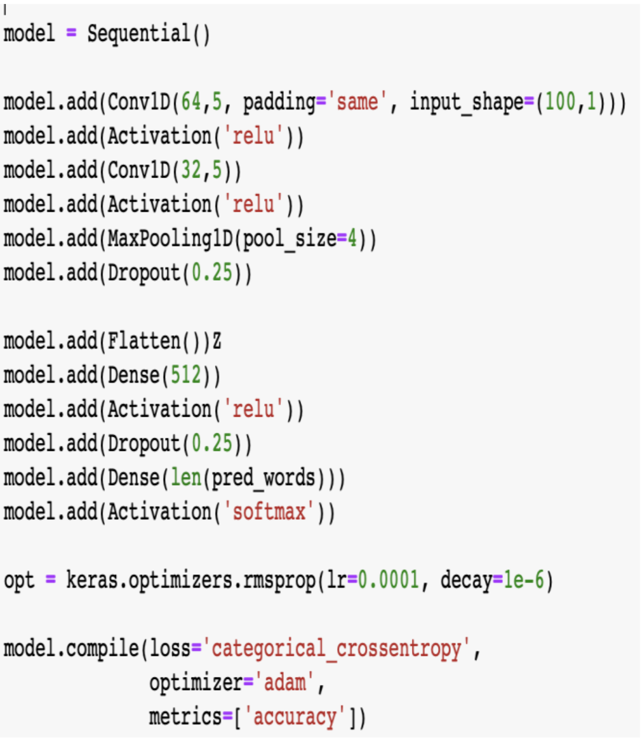

activation function
convolution
Sentence matrix
7 x 5
3 region sizes: (2,3,4)
2 filters for each region size
totally 6 filters
2 feature maps for each region size
1-max pooling
6 univariate vectors concatenated together to form a single feature vector
softmax function regularization in this layer
2 classes
d=5
I
like
this
movie
very
much
!

Figure 14: Architecture of Convolutional Neural Networks

5.1. Parameters used for our CNN:

In our CNN model we will be using two convolutional layers, for the first convolutional layer we have used 64 filters with each filter size of 4 we have followed up with relu activation combined with Max pool layer of size 2 after this for the second convolution layer we have used 32 filters of size 4 and followed up with relu activation combined with Max pool layer of size 2. High number of filters helps in increasing the learning rate. After second convolution layer we have used dense layer of size 512 with relu activation and for the final layer we have used a dense layer of size 4 with SoftMax activation. For the 2 convolution layers we have used dropout function to generalize the results. **(Kim, 2017)**

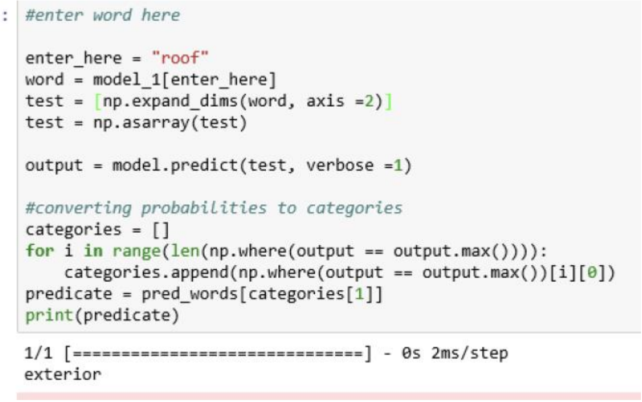


```
model = Sequential()

model.add(Conv1D(64,5, padding='same', input_shape=(100,1)))
model.add(Activation('relu'))
model.add(Conv1D(32,5))
model.add(Activation('relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.25))

model.add(Flatten())Z
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dropout(0.25))
model.add(Dense(len(pred_words)))
model.add(Activation('softmax'))

opt = keras.optimizers.rmsprop(lr=0.0001, decay=1e-6)

model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Snippet of code for the model

6 Model results

The trained model can now be used for predicting the predicate when given an object, which is a noun. The figures below will give an idea about the results provided by the CNN model.



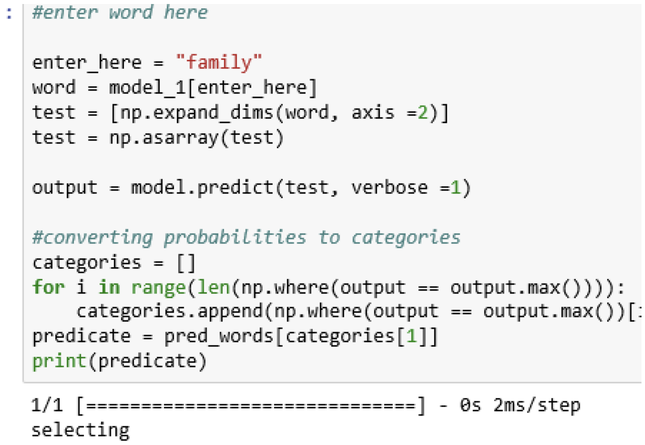
```
#enter word here

enter_here = "roof"
word = model_1[enter_here]
test = [np.expand_dims(word, axis =2)]
test = np.asarray(test)

output = model.predict(test, verbose =1)

#converting probabilities to categories
categories = []
for i in range(len(np.where(output == output.max()))):
    categories.append(np.where(output == output.max())[i][0])
predicate = pred_words[categories[1]]
print(predicate)
```

```
1/1 [==============================] - 0s 2ms/step
exterior
```

Figure 15: Snapshot of the model predicting the predicate for the word “Roof”.

In the figure 15 we can see that when the word roof is passes as a subject to the model the word exterior is generated as predicate.

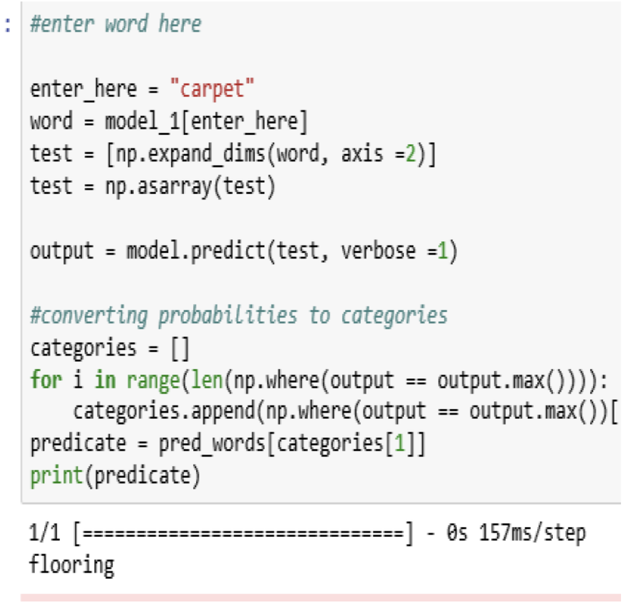
This predicate can be clearly used for finding the relation between the subject roof and other objects.



```
: #enter word here

  enter_here = "family"
  word = model_1[enter_here]
  test = [np.expand_dims(word, axis =2)]
  test = np.asarray(test)

  output = model.predict(test, verbose =1)

  #converting probabilities to categories
  categories = []
  for i in range(len(np.where(output == output.max()))):
      categories.append(np.where(output == output.max())[
  predicate = pred_words[categories[1]]
  print(predicate)

  1/1 [==============================] - 0s 2ms/step
  selecting
```

Figure 16: Snapshot of the model predicting the predicate for the word "family".

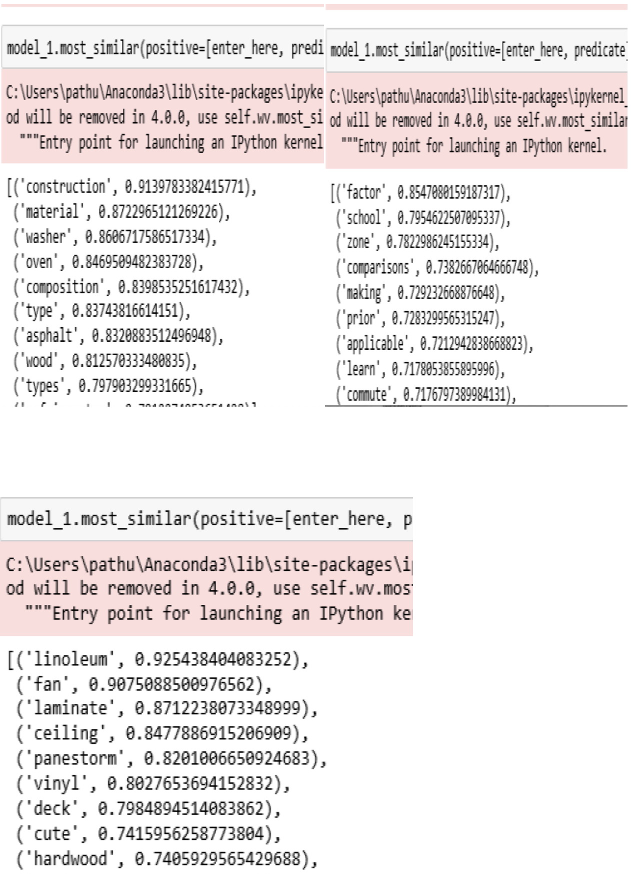
In the figure 16 we can see that when the word family is passed as a subject to the model the word selecting is generated as predicate. This predicate can be clearly used for finding the relation between the subject family and other objects.



```
: #enter word here

  enter_here = "carpet"
  word = model_1[enter_here]
  test = [np.expand_dims(word, axis =2)]
  test = np.asarray(test)

  output = model.predict(test, verbose =1)

  #converting probabilities to categories
  categories = []
  for i in range(len(np.where(output == output.max()))):
      categories.append(np.where(output == output.max())[
  predicate = pred_words[categories[1]]
  print(predicate)

  1/1 [==============================] - 0s 157ms/step
  flooring
```

Figure 17: Snapshot of the model predicting the predicate for the word "carpet".

In the figure 17 we can see that when the word carpet is passed as a subject to the model the word selecting is generated as predicate. This predicate can be clearly used for finding the relation between the subject carpet and other objects.



```
model_1.most_similar(positive=[enter_here, predi

C:\Users\pathu\Anaconda3\lib\site-packages\ipyke
od will be removed in 4.0.0, use self.wv.most_si
  """Entry point for launching an IPython kernel

[('construction', 0.9139783382415771),
 ('material', 0.8722965121269226),
 ('washer', 0.8606717586517334),
 ('oven', 0.8469509482383728),
 ('composition', 0.8398535251617432),
 ('type', 0.83743816614151),
 ('asphalt', 0.8320883512496948),
 ('wood', 0.812570333480835),
 ('types', 0.797903299331665),
```

```
model_1.most_similar(positive=[enter_here, predicate]

C:\Users\pathu\Anaconda3\lib\site-packages\ipykernel_
od will be removed in 4.0.0, use self.wv.most_similar
  """Entry point for launching an IPython kernel.

[('factor', 0.8547080159187317),
 ('school', 0.795462250709533 7),
 ('zone', 0.782298624515533 4),
 ('comparisons', 0.7382667064666748),
 ('making', 0.729232668876648),
 ('prior', 0.728329956531524 7),
 ('applicable', 0.7212942838668823),
 ('learn', 0.7178053855895996),
 ('commute', 0.717679738998413 1),
```

```
model_1.most_similar(positive=[enter_here, p

C:\Users\pathu\Anaconda3\lib\site-packages\i
od will be removed in 4.0.0, use self.wv.mos
  """Entry point for launching an IPython ke

[('linoleum', 0.9254384040832 52),
 ('fan', 0.9075088500976562),
 ('laminate', 0.8712238073348999),
 ('ceiling', 0.8477886915206909),
 ('panestorm', 0.8201006650924683),
 ('vinyl', 0.8027653694152832),
 ('deck', 0.7984894514083862),
 ('cute', 0.7415956258773804),
 ('hardwood', 0.7405929565429688),
```

Figure 17: the above figure shows the object using the most similar words for the predicates. 1) predicate for roof, 2) predicate for family, and 3) predicate for carpet

Though the goal for this project is to generate the predicate for the subject. We also went forward to generate the possible objects for the predicated generated in the previous steps.

Though we did not predict the exact objects for the predicates we have just generated the most possible objects. In the figure 17 we can see that the possible objects for the subject roof are material, asphalt and wood. Therefore, when we combine the subject, object and object we will get the relations like:

- "roof exterior asphalt"
- "roof exterior wood"

Ontology creation on unstructured data.

Another example is, in the figure 17 we can see that the possible objects for the subject family are school, commute and zone. Therefore, when we combine the subject, object and object we will get the relations like:

- “family selecting school”
- “family selecting commute”

We can also see that the possible objects for the subject carpet are linoleum, hardwood and vinyl. Therefore, when we combine the subject, object and object we will get the relations like:

- “carpet flooring vinyl”
- “carpet flooring hardwood”

When we look at these examples of relationship between the subjects and object like the relation between the roof and wood. This clearly shows that the roof has an exterior of wood.

In the next example, we can see the relation between family and school, commute. This shows that the families look for school and commute when they are looking for homes.

7 Role of each team member

Sooraj Cheeti: Worked on analyzing the data and the cleaning the data.

Dilip Molugu: worked on cleaning the data and creating the data that can be used for training data for the CNN.

Nikhil Reddy Pathuri: worked on creating and optimizing the CNN model for predicting the predicate.

8 Conclusion

This technique can be used to develop the ontologies only based on the unstructured data. This helps in automatic generation of the ontologies without using any tools like protégé which requires manual creation of ontologies.

As the data used in this project is limited and the results produced are not very accurate in some situations, if the size of the data is more in a given domain and the model can be trained in a better way to get more accurate results. Therefore, accuracy can also be increased with increase in the size of data.

We have used all the data provided in the raw Zillow housing data. Also, the accuracy of the model can be further increased by only considering the reviews provided by the users. The reason for this is that the reviews are written by the users and do not contain any markup language tags. This will make the prediction more accurate.

References

CHia, D. (2018, 12 5). *A line-by-line implementation guide to Word2Vec using Numpy*. Retrieved from Towards Data Science: https://towardsdatascience.com/an-implementation-guide-to-word2vec-using-numpy-and-google-sheets-13445eebd281

Fortney, K. (2017, 11 28). *Pre-Processing in Natural Language Machine Learning*. Retrieved from Towards Data Science: https://towardsdatascience.com/pre-processing-in-natural-language-machine-learning-898a84b8bd47

Kim, J. (2017, 12 7). *Understanding how Convolutional Neural Network (CNN) perform text classification with word embeddings*. Retrieved from joshuakim: http://www.joshuakim.io/understanding-how-convolutional-neural-network-cnn-perform-text-classification-with-word-embeddings/

Novelli, A. D., & Parante de Oliveria, J. M. (2012). Simple Method for Ontology Automatic Extraction. *IJACSA*, 8.

Piantadosi, S. T. (2015, 10 1). *Zipf's word frequency law in natural language: A critical review and future directions*. Retrieved from NCBI: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4176592/

Survey Paper Question:

Paper1: A Lightweight Approach for Evaluating Sufficiency of Ontologies -By Lalit Mohan S, Gollapudi VRJ Sai Prasad, Sridhar Chimalakonda, Y. Raghu Reddy and Venkatesh Choppella.

This paper helped us in preparing the data required to train our CNN model.

1) **Pre-Processing:**

We first pre-processed our text as given in the paper by Lemmatizing the data and removing the stop-words like ‘and’, ‘but’, ‘if’, etc. Then we performed Part-of-Speech (POS) tagging to list Nouns and Verbs in the document.

2) **Text and Labels Extraction:**

Created a list containing clusters of words for the ontology by selecting the Nouns and Verbs which are obtained by POS Tagging. We then use Word2Vec and identify most similar and identify the synonyms for the extracted words. Then we Identified the most appropriate verbs associated to that noun and used those verbs as Labels.

Paper 2): Ontology Construction Based on Deep Learning -By Jianan Wang, Jin Liu, and Lei Kong.

The base idea of our project is based on this paper. This paper helped us in building the CNN model using a large amount of unstructured text and building the Ontology.

1) **Text to Vectors:**

This Paper helped us in preparing the pre-processed data required to train the CNN model. We needed to convert text to vectors that will be used as features by the CNN model. The paper presents a method for creating these vectors using the Word2Vec functionality.

2) **Convolution Neural Network (CNN) Model:**

We were able to learn about what is a CNN model and the steps to implement this model. The paper explained the different layers used in the CNN model and other tuning parameters like number of filters, filter size, Max-overtime pool size and the SoftMax Output. It explained the functionality of each layer, its use and the results obtained from each layer. We were able to understand a clear idea of the CNN Architecture and used this knowledge to implement best parameter settings required to train our model.

The labels obtained with the help of paper 1 are used to create vectors and used these vectors as training data to the CNN model which is built based on the steps mentioned in paper 2.