Project 4 Report - CMSC 409 - Artificial Intelligence

Steven Hernandez

In total, there are 196 unique root words found.

24 words that are encountered at least 2 times.

And then only 18 words that are encountered at least 3 times.

These statistics are calculated based on processing the documents in the following ways:

a. Tokenizing the sentences, which splits each sentence on the spaces to only produce a list of word/numeric tokens. This allows us to begin processing each word individually without requiring the context of the entire sentence.
b. Removing punctuation is required because in general, punctuation does not provide us textual context. Again, we are only looking at the similarity of sentence based on the number of occurrences of common words between the sentences. We are not trying to decifer the intent or the sentiment behind the sentence, so we do not require punctuation or even placement of words within the sentence. Just that the word exists within the sentence.
c. Removing numbers because numbers do not provide context about what the sentence is talking about. A book might cost $20 as would a basic microcontroller like an Arduino, but they are not related. Additional since, we removed punctuation in the previous step, we wouldn't be able to differentiate $20 from 20 miles or 20 participants, etc.
d. Converting upper to lower case prevents words appearing at the beginning of a sentence (with a required capital letter) from being considered a different word if it also appears in the middle of a sentence (which would be written in all lower case)
e. Removing stop words shrinks the total number of words that we find. More importantly though, it removes overly common words that do not provide us useful insights into the similarity of sentences. The word 'the' is very likely to appear in most sentences, thus is not a useful indicator.
f. Stemming takes a word in past tense/future tense or plural/singular and takes the 'stem' or 'root' word. This further shrinks the overall number of words or dimensions that we must analyze. An example: run and running have the same root word, thus are very similar.
g. Combining stemmed words takes these common stemmed root words and combines them so that we can get a total count of the occurances of the word throughout all sentence documents.

On the following page is a table listing all of these root words along with the number of occurrences of the word through the documents (the feature vector)

Root Word	# of instances	Root Word	# of instances	Root Word	# of instances	Root Word	# of instances
autonom	5	musk	1	public	1	reason	1
sedan	3	wai	2	us	1	attain	1
travel	1	escap	1	driven	1	multipl	1
type	1	obsolesc	1	combin	1	domain	1
road	5	have	1	two	2	feel	1
speed	1	sort	1	paradigm	1	express	1
up	2	merger	1	everyth	1	understand	2
mile	10	biolog	1	know	1	emot	1
per	7	entir	1	realiti	1	singl	1
hour	6	interior	1	sens	2	famili	1
futur	1	freshli	1	knowledg	2	conveni	1
machin	6	paint	1	experi	1	locat	1
learn	5	go	1	via	1	near	1
rai	2	percent	1	five	1	major	1
kurzweil	2	befor	1	on	2	rout	1
predict	1	lap	2	suit	1	author	1
year	1	possibl	2	king	1	book	1
awai	1	lead	2	bed	1	ag	1
singular	1	necessarili	1	veri	2	spiritu	1
selfimprov	1	sentienc	1	nice	1	describ	1
artifici	6	applianc	2	queen	1	spread	1
superintellig	1	includ	2	complet	1	throughout	1
far	1	well	1	over	2	cosmo	1
exce	1	secur	1	drive	2	cute	1
human	2	system	1	accidentfre	1	classi	1
intellig	11	tenant	1	updat	3	open	1
get	1	respons	1	renov	1	area	1
car	5	electr	1	new	1	great	1
kilomet	6	water	2	floor	1	went	1
second	2	ga	1	john	1	round	1
newli	1	pet	2	mccarthi	1	minut	1
remodel	1	negoti	1	inventor	1	recent	1
home	4	base	1	program	2	work	1
rent	1	anim	1	languag	1	fundament	1
bedroom	7	four	1	lisp	1	techniqu	1
bath	3	row	1	coin	1	deep	1
live	4	hous	3	term	1	lai	1
room	4	come	3	deal	1	groundwork	1
larg	3	washer	1	number	1	automat	1
eat	2	dryer	1	situat	1	through	1
kitchen	5	finish	1	averag	2	increas	1
full	1	basement	1	gallon	1	world	1
size	4	three	1	pound	1	bathroom	1
util	1	park	1	gener	1	townhous	1
test	2	space	1	t	1	central	1
achiev	1	back	1	selfawar	1	heat	1
rang	1	approv	1	comput	2	air	1
around	4	owner	1	engag	1	trash	1
charg	2	limit	1	commonsens	1	sewag	1

The following lists the root words with greater than 2occurrences:

Root Word	# of instances
autonom	5
sedan	3
road	5
mile	10
per	7
hour	6
machin	6
learn	5
artifici	6
intellig	11
car	5
kilomet	6
home	4
bedroom	7
bath	3
live	4
room	4
larg	3
kitchen	5
size	4
around	4
hous	3
come	3
updat	3

The following 2 tables show the distribution of root words which appear at least 2 times across each document (with each row indicating one sentence) (This is the Term Document Matrix **TDM**)

sentence #	autonom	sedan	road	mile	per	hour	machin	learn	artifici	intellig	car	kilomet
1	1	1	1	1	1	1	0	0	0	0	0	0
2	0	0	0	0	0	0	1	1	1	1	0	0
3	0	0	0	1	1	1	0	0	0	0	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	1	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1	0	0	1	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0	0	1	1
9	1	1	0	1	1	1	0	0	0	0	0	1
10	0	0	0	0	0	0	1	1	1	1	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0
14	1	0	1	0	0	0	0	0	0	0	1	1
15	0	0	0	0	0	0	0	0	1	1	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0
17	1	0	0	1	0	0	0	0	0	0	0	1
18	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	1	1	0	0
20	1	0	1	0	0	0	0	0	0	0	1	0
21	0	1	0	1	1	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	1	1	1	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	1	0	0	1	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	1	1	1	1	0	0	0	0	1	0
27	0	0	0	0	0	0	1	1	1	1	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0

sentence #	bedroom	bath	live	room	larg	kitchen	size	around	hous	come	updat
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	1	1	1	1	0	0	0	0
5	0	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	1	0	1	1	1	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0
12	1	1	0	0	0	0	0	0	1	1	0
13	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	1	0
16	1	0	0	0	1	0	1	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	1	0	0	1	0	1
19	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0
23	1	1	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0
25	0	0	1	0	0	1	0	0	1	0	1
26	0	0	0	0	0	0	0	1	0	0	0
27	0	0	0	0	0	0	0	1	0	0	0
28	1	0	1	1	0	1	0	0	0	0	0

Learning

We begin learning by using the 'Winner Takes All' (WTA) method which means that we begin with `n` clusters, then iterating for each document, we find the closest cluster using euclidean distance. Depending on which cluster's center (based on weight) is closest to the new document, the cluster's center's weight is changed by a value to better match the resulting pattern. Code below:

```
def learn_wta(data, cluster_count=1):
    seed = 9
    # Create randomly initialized weights for the number of expected clusters
    # Each of these weight sets with len(data[0]) number of weights
    weights = np.random.rand(cluster_count, len(data[0]))

    for i in range(1000):
        # shuffle(data)
        for pattern in data:
            index = get_closest_cluster(weights, pattern)
            weights[index] += calculate_change_in_weight(weights[index], pattern)

    return weights


def get_closest_cluster(cluster_weights, current_pattern):
    cluster_distances = np.zeros(len(cluster_weights))

    # Euclidean distance:

    # For each cluster
    for i in range(len(cluster_weights)):
        cluster_distances[i] = 0

        # For each word
        for j in range(len(current_pattern)):
            val = pow((current_pattern[j] - cluster_weights[i][j]), 2)
            cluster_distances[i] += val

    least_distance_index = 0

    for i in range(len(cluster_distances)):
        if cluster_distances[i] < cluster_distances[least_distance_index]:
            least_distance_index = i

    return least_distance_index


def calculate_change_in_weight(current_weights, current_pattern):
    epsilon = 0.05
    return epsilon * (current_pattern - current_weights)
```

Learned clusters:

Cluster 1:

- 11) Four bedroom 3 bath row house, home comes with 2 washers and 2 dryers and finished basement.
- 22) Single family home with 5 bedroom and 2.5 bath, conveniently located near all major routes.

Cluster 2:

- 0) The autonomous sedan will be able to travel on any type of road at speeds of up to 60 miles per hour.
- 8) The autonomous sedan will do a lap or 2 at around 250 kilometers per hour (149 miles per hour).

Cluster 3:

- 3) Newly remodeled home for rent, 4 bedrooms with 1 bath, living room, large eat in kitchen with a full sized utility room.
- 6) Entire interior of home is freshly painted, large living room and bedroom.
- 15) One of the bedroom is a large suit with a king size bed, the other one is a very nice size bedroom with a queen size.
- 27) Two bedroom 1 bathroom townhouse, central heat and air, water trash sewage included, living room, eat in kitchen.

Cluster 4:

- 2) This gets the car from 0 to 60 miles per hour (that is, to 97 kilometers per hour) in 3.2 seconds.
- 4) On the road test, we were able to achieve a range of 220 kilometers (around 138 miles) on a charge.
- 7) The car will go 443 kilometers (275 miles) on a charge, up 3.7 percent from before.
- 10) All appliances are included, as well as security system, tenant is responsible for electric and water gas, pets negotiable based on animal.
- 12) Three parking spaces in back, pets are possible with approval from the owner.
- 13) The 31.5 kilometers of roads that are off limits to the public, will be used for testing the autonomously driven car.
- 16) They have completed over 300,000 autonomous driving miles (500,000 kilometers) accident-free.
- 17) House has been updated and renovated with an updated kitchen and new flooring.
- 19) The car had to autonomously deal with a number of situations on the road.
- 20) Over 839 miles of driving, we averaged 29 miles per gallon, for the 3317 pound sedan.
- 24) Very cute and classy house with open living area and kitchen, kitchen is updated with great appliances.
- 25) The car went round the 3 mile off road lap in 11 minutes and 50 seconds, which is an average of around 15 miles per hour.

Cluster 5:

- 1) On future of machine learning, Ray Kurzweil has predicted that we are only 28 years away from the Singularity or when self-improving artificial super-intelligence will far exceed human intelligence.
- 5) Musk said that the way to escape human obsolescence may be by having some sort of merger of biological intelligence and machine intelligence.

- 9) While artificial intelligence could possibly lead to intelligence in machines with machine learning, intelligence will not necessarily lead to sentience.
- 14) Artificial intelligence is combining two paradigms, that everything that we know about our reality comes by way of our senses, and that the knowledge comes from our experiences via five senses.
- 18) John McCarthy, inventor of the programming language LISP, coined the term "artificial intelligence" in 1955.
- 21) General Artificial Intelligence -t he self-aware computer programs that can engage in commonsense reasoning and learning, attain knowledge in multiple domains, feel, express and understand emotions.
- 23) Ray Kurzweil, author of the 1999 book The Age of Spiritual Machines described that intelligence would spread throughout the cosmos.
- 26) Recent artificial intelligence work has been fundamental, with techniques like deep learning laying the groundwork for computers that can automatically through machine learning increase their understanding of the world around them.

If we look at the feature vectors as a bit map showing whether a sentence has or does not have a specific word, we can begin to see the pattern of the clustering method.

Cluster 1:

- 11) 000000000000111000000110
- 22) 000000000000111000000000

Cluster 2:

- 0) 111111000000000000000000
- 8) 110111000001000000001000

Cluster 3:

- 3) 000000000000111111110000
- 6) 000000000000110111000000
- 15) 000000000000010001010000
- 27) 000000000000010110100000

Cluster 4:

- 2) 000111000011000000000000
- 4) 001100000001000000001000
- 7) 000100000011000000000000
- 10) 000000000000000000000000
- 12) 000000000000000000000000
- 13) 101000000011000000000000
- 16) 100100000001000000000000
- 17) 000000000000000000100101
- 19) 101000000010000000000000
- 20) 010110000000000000000000
- 24) 000000000000000100100101

- 25) 00111100001000000000001000

Cluster 5:

- 1) 00000011110000000000000
- 5) 00000010010000000000000
- 9) 00000011110000000000000
- 14) 00000000110000000000010
- 18) 00000000110000000000000
- 21) 00000001110000000000000
- 23) 00000010010000000000000
- 26) 00000011110000000001000

From these bit maps, we can see that each cluster has relatively distinct columns which matchacross the documents of the cluster.

Of course, this clustering does split some groups of documents into more clusters than expected. Some clusters seem as if they could be combined to the human views. Having additional sample documents would very likely help with this issue. With these few number of documents, for example, sentence 12 'Three parking spaces in back, pets are possible with approval from the owner.' does not mention being about a 'home' or many other words which are used in other documents that truly identify itas being about a home. With more documents, we would begin to have more overlap, which could aid in finding which words provide us the most importance. Sentence 10 as well does not share enoughwords to be able to identify it with the provided documents.

Below, we can see which words these sentences share in common.

Cluster 1:

- 11) home, bedroom, bath, hous, come
- 22) home, bedroom, bath

Cluster 2:

- 0) autonom, sedan, road, mile, per, hour
- 8) autonom, sedan, mile, per, hour, kilomet, around

Cluster 3:

- 3) home, bedroom, bath, live, room, larg, kitchen, size
- 6) home, bedroom, live, room, larg
- 15) bedroom, larg, size
- 27) bedroom, live, room, kitchen

Cluster 4:

- 2) mile, per, hour, car, kilomet
- 4) road, mile, kilomet, around
- 7) mile, car, kilomet
- 10)
- 12)
- 13) autonom, road, car, kilomet

- 16) autonom, mile, kilomet
- 17) kitchen, hous, updat
- 19) autonom, road, car
- 20) sedan, mile, per
- 24) live, kitchen, hous, updat
- 25) road, mile, per, hour, car, around

Cluster 5:

- 1) machin, learn, artifici, intellig
- 5) machin, intellig
- 9) machin, learn, artifici, intellig
- 14) artifici, intellig, come
- 18) artifici, intellig
- 21) learn, artifici, intellig
- 23) machin, intellig
- 26) machin, learn, artifici, intellig, around

One problem of this method compared to a method where clusters a created as needed, was that if the random initialization of weights for the cluster were randomly generated in a bad spot, it is likely the cluster would never contain any sentences because (as the name implies) the Winner Takes All methodwould often find one cluster taking over most of the documents, while other clusters remained empty.

The solution taken here for this problem was to learn on many randomly placed clusters. Learning began with 20 clusters. From these 20 clusters however, we only end up with 5 clusters. Additionally, (during testing) it would some times result in clusters with only a single result, when the result would have worked better in some other already defined cluster.

With fewer clusters (for example 4), we occasionally ended up with good results, but often would end upwith most documents stuck in one single cluster

In addition to having more documents to sample, having clusters only as needed would likely improve this situation. With clusters-as-needed, clusters would only be able to contain documents within some radius of the cluster's center. If a document is found outside of this radius, then a new cluster would be formed in this place.