Introduction to Text Feature Engineering

Text Feature Engineering

- Machine Learning algorithms (Almost all) cannot accept text as input
- **Text Feature Engineering:** Convert text to features

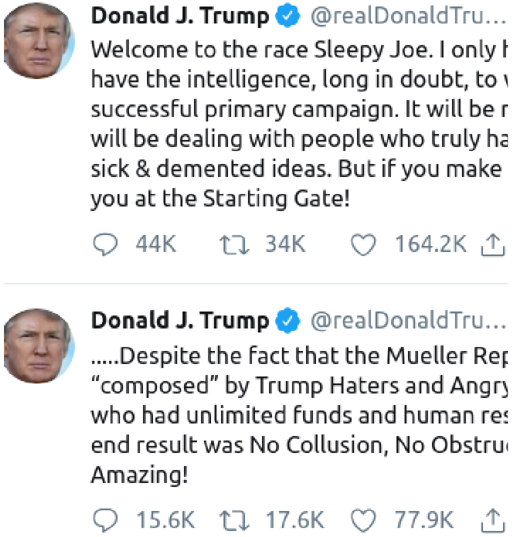
Text Data

Movie Reviews

Text Data

Movie Reviews

Tweets



Analytics Vidhya
Learn everything about analytics

Text Data

Movie Reviews


Fight Club (1999)
User Reviews
Review this title

3,356 Reviews
Hide Spoilers Filter by Rating: Show All

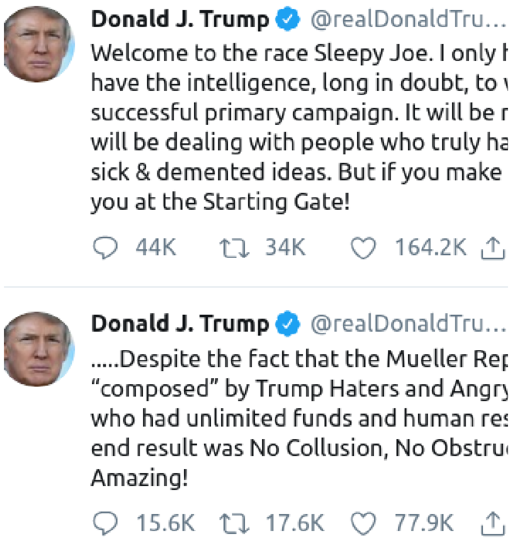
10/10

A unique film
buk-3 15 October 1999

Fight Club is one of the most unique films I have ever seen. I
take on life, FC also presents its material in a fresh way. My r
opinion, it does not present characters for us to think about. F
about. I will say that I cannot recall *ever* having been "asked
the way this film asks in its third act AND at the same time co
there is no room--or need--for disbelief.

Perhaps these comments will not make sense to the average
and, unfortunately, its premise--as another hollywood flick fille
as to say that this film is not about violence. It is about choice
It is about waking up and realizing that at some point in the p
our dreams without even realizing that society has stuck its fi

Tweets

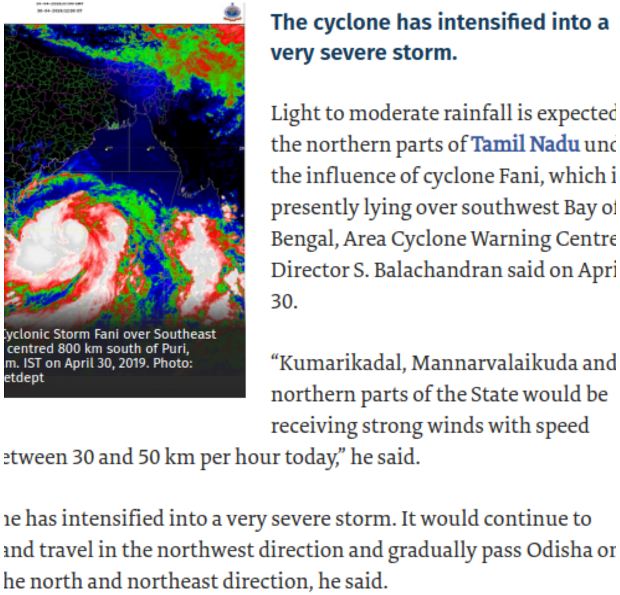

Donald J. Trump @realDonaldTru…
Welcome to the race Sleepy Joe. I only h
have the intelligence, long in doubt, to w
successful primary campaign. It will be n
will be dealing with people who truly ha
sick & demented ideas. But if you make
you at the Starting Gate!

44K 34K 164.2K

Donald J. Trump @realDonaldTru…
.....Despite the fact that the Mueller Rep
"composed" by Trump Haters and Angry
who had unlimited funds and human res
end result was No Collusion, No Obstru
Amazing!

15.6K 17.6K 77.9K

Online News


The cyclone has intensified into a very severe storm.

Light to moderate rainfall is expected
the northern parts of Tamil Nadu und
the influence of cyclone Fani, which i
presently lying over southwest Bay of
Bengal, Area Cyclone Warning Centre
Director S. Balachandran said on Apri
30.

"Kumarikadal, Mannarvalaikuda and
northern parts of the State would be
receiving strong winds with speed
etween 30 and 50 km per hour today," he said.

ie has intensified into a very severe storm. It would continue to
and travel in the northwest direction and gradually pass Odisha or
he north and northeast direction, he said.

yclonic Storm Fani over Southeast
centred 800 km south of Puri,
m. IST on April 30, 2019. Photo:
etdept

Analytics Vidhya
Learn everything about analytics

Text Feature Engineering

- ❑ Machine Learning algorithms (Almost all) cannot accept text as input
- ❑ **Text Feature Engineering:** Convert text to features
- ❑ Information in text is vital

Text Feature Engineering

- Machine Learning algorithms (Almost all) cannot accept text as input
- **Text Feature Engineering:** Convert text to features
- Information in text is vital
    - word-count, character-count, negation word-count etc.

Text Feature Engineering

- Machine Learning algorithms (Almost all) cannot accept text as input
- **Text Feature Engineering:** Convert text to features
- Information in text is vital
  - word-count, character-count, negation word-count etc.
  - dates, emails, phone numbers

Text Feature Engineering

- Machine Learning algorithms (Almost all) cannot accept text as input
- **Text Feature Engineering:** Convert text to features
- Information in text is vital
    - word-count, character-count, negation word-count etc.
    - dates, emails, phone numbers
    - Sentiment: positive, negative, or neutral

Understanding Regular Expressions (RegEx)

What are Regular Expressions?

Name
Sunil
Sumit
Ankit
Surjeet
Surabhi

What are Regular Expressions?

Name
Sunil
Sumit
Ankit
Surjeet
Surabhi

S u _ _ _

Find the names that fit the pattern above.

What are Regular Expressions?

Name
Sunil
Sumit
Ankit
Surjeet
Surabhi

S u _ _ _

Find the names that fit the pattern above.

Analytics Vidhya
Learn everything about analytics

What are Regular Expressions?

- ❑ Patterns special characters having an associated textual meaning (ex: “\d” : “numbers”)

```
John's Salary is $5000, he lives in the block 5 of the 3rd
Manhatten street. He was born in the year 1990.
```

- ❑ Used for writing rule-based information mining systems
- ❑ RegEx can be used for text cleaning also

Why Text Cleaning is Required?

When I pay as much as I do for a #phone I expect it to work. \n A very unhappy %@#& customer.

Why Text Cleaning is Required?

When I pay as much as I do for a #phone I expect it to work. \n A very unhappy %@#& customer.

[“When”, “I”, “pay”, “as”, “much”, “as”, “I”, “do”, “for”, “a”, “#phone”, “I”, “expect”, “it”, “to”, “work.”, “\n”, “A”, “very”, “unhappy”, “%@#&”, “customer.”]

Creating Linguistic Features

Linguistic Features

Amazon is working on a device that can read emotions

Linguistic Features

Amazon is working on a device that can read emotions

Nouns: 3

Linguistic Features

Amazon is working on a device that can read emotions

Nouns: 3

Verbs: 4

Part of Speech Tagging

Part of Speech Tagging

- Defines the syntactic context and role of words in the sentence

Part of Speech Tagging

- ❑ Defines the syntactic context and role of words in the sentence.
- ❑ Common POS Tags : Nouns, Verbs, Adjectives, Adverbs

Part of Speech Tagging

- Defines the syntactic context and role of words in the sentence.
- Common POS Tags : Nouns, Verbs, Adjectives, Adverbs

Sentence : David has purchased a new Laptop from Apple Store

Part of Speech Tagging

- Defines the syntactic context and role of words in the sentence.
- Common POS Tags : Nouns, Verbs, Adjectives, Adverbs

Sentence : David has purchased a new Laptop from Apple Store

Analytics Vidhya
Learn everything about analytics

Part of Speech Tagging

- Defines the syntactic context and role of words in the sentence.
- Common POS Tags : Nouns, Verbs, Adjectives, Adverbs

Sentence : David has purchased a new Laptop from Apple Store

- Defined by their relationship with the adjacent words

Part of Speech Tagging

- **spaCy** or **NLTK** can be used for POS tagging
- spaCy is more advanced and feature rich

Creating Bag of Words Features

Bag of Words

I love playing guitar					
I hate playing guitar					

Bag of Words

	I	love	playing	guitar	hate
I love playing guitar					
I hate playing guitar					

Bag of Words

Vocabulary

	I	love	playing	guitar	hate
I love playing guitar					
I hate playing guitar					

Bag of Words

Vocabulary

	I	love	playin g	guita r	hat e
I love playing guitar	?	?	?	?	?
I hate playing guitar	?	?	?	?	?

Analytics Vidhya
Learn everything about analytics

Bag of Words

	I	love	playing	guitar	hate
I love playing guitar	1	1	1	1	0
I hate playing guitar					

Bag of Words

	I	love	playing	guitar	hate
I love playing guitar	1	1	1	1	0
I hate playing guitar	1	0	1	1	1

Bag of Words - Challenges

- High dimensionality

  Vocabulary = Dimensions

- Same words with different meanings

  “He is the **right** man for the position”

  “Everyone has the **right** to freedom of opinion and expression"

Text Pre-processing

Text Pre-processing

Always end day with positive thought matter hard things Tomorrow's fresh opportunity make better

Text Pre-processing

Always end the day with a positive thought. No matter how hard things were. Tomorrow’s a fresh opportunity to make it better.

What are Stop Words?

What are Stop Words?

- Extremely common but of little value

a	an	and	are	as	at	be	by	for	from
has	he	in	is	it	its	of	on	that	the
to	was	were	will	with					

- Removing stop words reduces vocabulary size

Remove Stopwords

- ❑ Consider the sentences below

1. “Sam waited for the train”
2. “the train was late”

Remove Stopwords

- Consider the sentences below

	Sam	waited	for	the	train	was	late
Sam waited for the train	1	1	1	1	1	0	0
the train was late	0	0	0	1	1	1	1

Remove Stopwords

- Consider the sentences below

	Sam	**waited**	**for**	**the**	**train**	**was**	**late**
Sam waited ~~for~~ ~~the~~ train	1	1	1	1	1	0	0
~~the~~ train ~~was~~ late	0	0	0	1	1	1	1

Remove Stopwords

- Consider the sentences below

	Sam	waited	train	late
Sam waited ~~for~~ ~~the~~ train	1	1	1	0
~~the~~ train ~~was~~ late	0	0	1	1

Text Normalization

- **Morpheme:** base form of word

Text Normalization

- **Morpheme:** base form of word
- Structure of token : <prefix> <morpheme> <suffix>

Text Normalization

- **Morpheme:** base form of word
- Structure of token : <prefix> <morpheme> <suffix>

  Example: Antinationalist = Anti + national + ist

Text Normalization

- **Morpheme:** base form of word
- Structure of token : <prefix> <morpheme> <suffix>

  Example: Antinationalist = Anti + national + ist
- **Normalization:** Process of converting a token into its base form (morpheme)

Text Normalization

- **Morpheme:** base form of word
- Structure of token : <prefix> <morpheme> <suffix>

  Example: Antinationalist = Anti + national + ist
- **Normalization:** Process of converting a token into its base form (morpheme)
- Types: **Stemming** and **Lemmatization**

Text Normalization: Stemming

- Elementary rule based process to remove inflectional forms from a token.

Text Normalization: Stemming

- Elementary rule based process to remove inflectional forms from a token.
- “laughing”, “laughs”, “laugh”, “laughed”

Text Normalization: Stemming

- Elementary rule based process to remove inflectional forms from a token.
- “laughing”, “laughs”, “laugh”, “laughed” >> laugh

Text Normalization: Stemming

- Elementary rule based process to remove inflectional forms from a token.
- “laughing”, “laughs”, “laugh”, “laughed” >> laugh
- May generate non-meaningful terms

Text Normalization: Stemming

- Elementary rule based process to remove inflectional forms from a token.
- “laughing”, “laughs”, “laugh”, “laughed” >> laugh
- May generate non-meaningful terms

“his teams are not winning”

Text Normalization: Stemming

- Elementary rule based process to remove inflectional forms from a token.
- “laughing”, “laughs”, “laugh”, “laughed” >> laugh
- May generate non-meaningful terms

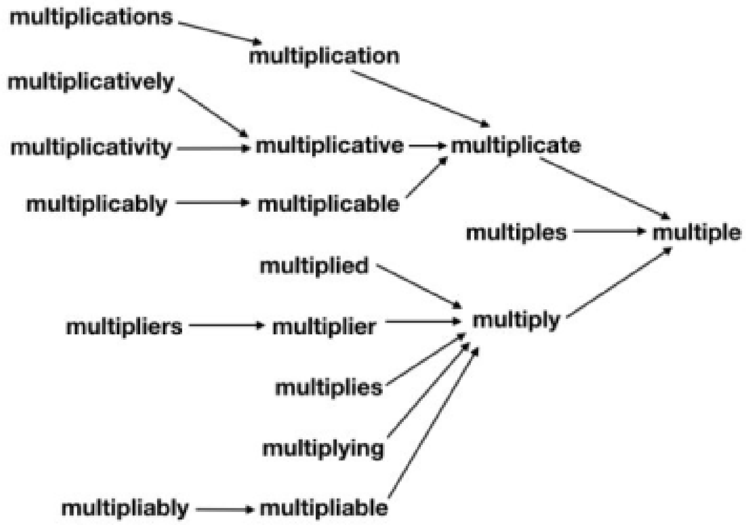
  “his teams are not winning” >> “hi team are not winn”

Text Normalization: Lemmatization

- Systematic process for reducing a token to its lemma

Text Normalization: Lemmatization

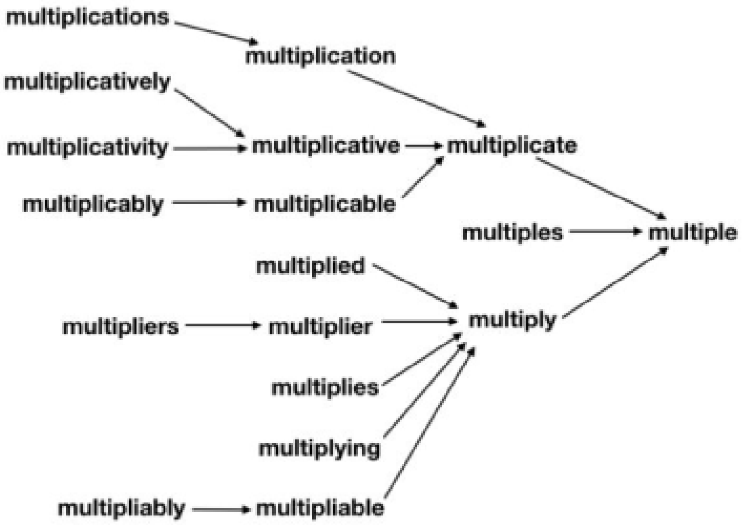
- Systematic process for reducing a token to its lemma
- Makes use of vocabulary, word-structure, part-of-speech tags and grammar relations



Analytics Vidhya
Learn everything about analytics

Text Normalization: Lemmatization

- Systematic process for reducing a token to its lemma
- Makes use of vocabulary, word-structure, part-of-speech tags and grammar relations
- Example: am, are, is >> be
  running, ran, runs >> run



Analytics Vidhya
Learn everything about analytics

Creating TF-IDF Features

Term Frequency and Inverse Document Frequency

- **TF (Term Frequency):** Frequency of a token in a document

- **IDF (Inverse Document Frequency):** Number of documents in which a specific term appears

Term Frequency and Inverse Document Frequency

- **TF (Term Frequency):** Frequency of a token in a document
- **IDF (Inverse Document Frequency):** Number of documents in which a specific term appears

$$\textbf{Term Frequency} = \frac{\text{Count of term } \mathbf{i} \text{ in a document } \mathbf{j}}{\text{number of terms in document } \mathbf{j}}$$

Term Frequency and Inverse Document Frequency

- **TF (Term Frequency):** Frequency of a token in a document
- **IDF (Inverse Document Frequency):** Number of documents in which a specific term appears

$$\textbf{Term Frequency} = \frac{\text{Count of term } \mathbf{i} \text{ in a document } \mathbf{j}}{\text{number of terms in document } \mathbf{j}}$$

$$\textbf{Inverse Document Frequency} = \log\left(\frac{\text{Count of documents in corpus}}{\text{Count of documents carrying term } \mathbf{i}}\right)$$

Analytics Vidhya
Learn everything about analytics

TF-IDF Score

TF-IDF Score = Term Frequency * Inverse Document Frequency

TF-IDF Score: Example

Corpus: 10,000 documents

TF-IDF Score: Example

Corpus: 10,000 documents

Terms: Delhi, Mumbai, Chennai

TF-IDF Score: Example

Corpus: 10,000 documents

Terms: Delhi, Mumbai, Chennai

City	Docs Count
Delhi	50
Mumbai	1300
Chennai	250

TF-IDF Score: Example

Corpus: 10,000 documents

Terms: Delhi, Mumbai, Chennai

New Document (total terms = 20)

City	Term Count
Delhi	3
Mumbai	2
Chennai	1

City	**Docs Count**
Delhi	50
Mumbai	1300
Chennai	250

TF-IDF Score: Example

Corpus: 10,000 documents

Terms: Delhi, Mumbai, Chennai

New Document (total terms = 20)

City	Term Count	TF
Delhi	3	3 / 20
Mumbai	2	2 / 20
Chennai	1	1 / 20

City	**Docs Count**
Delhi	50
Mumbai	1300
Chennai	250

TF-IDF Score: Example

Corpus: 10,000 documents

Terms: Delhi, Mumbai, Chennai

New Document (total terms = 20)

City	**Docs Count**
Delhi	50
Mumbai	1300
Chennai	250

City	Term Count	TF	IDF	TF-IDF
Delhi	3	3 / 20	$\log(10^4/ 50)$	
Mumbai	2	2 / 20	$\log(10^4/ 1300)$	
Chennai	1	1 / 20	$\log(10^4/ 250)$	

TF-IDF Score: Example

Corpus: 10,000 documents

Terms: Delhi, Mumbai, Chennai

New Document (total terms = 20)

City	**Docs Count**
Delhi	50
Mumbai	1300
Chennai	250

City	Term Count	TF	IDF	TF-IDF
Delhi	3	0.15	2.3	0.35
Mumbai	2	0.1	0.89	0.09
Chennai	1	0.05	1.6	0.08

TF-IDF Score

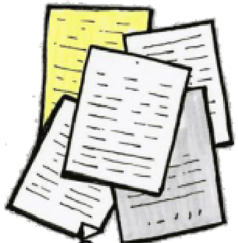
- TF-IDF score is high for terms which appears quite often in a document but are not present in most of the other documents.

- TF-IDF score is lower for terms which are occurring frequently in most of the documents in a corpus.

  Example - Stop words (“is”, “the”, “a”, “of”, etc.)

Word Embeddings

Word Embeddings

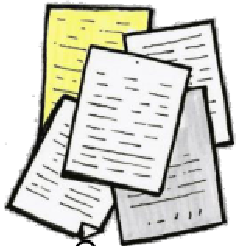


Corpus

```
array([[-0.01236233, -0.04655259,  0.00508882, ..., -0.00993368,
         0.01379246,  0.00122126],
       [-0.03087116, -0.02232517,  0.01138248, ..., -0.02389362,
         0.02484551, -0.0087585 ],
       [-0.03504547, -0.04104917,  0.00930308, ..., -0.03002032,
         0.01539359, -0.00338876],
       ...,
       [-0.03802555, -0.017358  ,  0.02445563, ..., -0.0131221 ,
         0.02305542, -0.00747857],
       [-0.02819404, -0.04432267,  0.01159158, ..., -0.02953893,
         0.01612862, -0.0099255 ],
       [-0.0326709 , -0.0484228 ,  0.01606839, ..., -0.03584684,
         0.00761068, -0.00948259]], dtype=float32)
```

Word Vectors

Word Embeddings

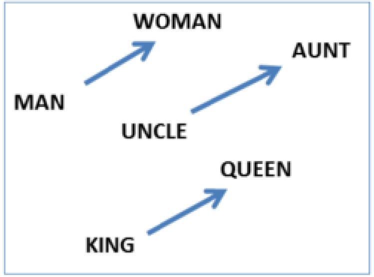


Corpus

```
array([[-0.01236233, -0.04655259,  0.00508882, ..., -0.00993368,
         0.01379246,  0.00122126],
       [-0.03087116, -0.02232517,  0.01138248, ..., -0.02389362,
         0.02484551, -0.0087585 ],
       [-0.03504547, -0.04104917,  0.00930308, ..., -0.03002032,
         0.01539359, -0.00338876],
       ...,
       [-0.03802555, -0.017358  ,  0.02445563, ..., -0.0131221 ,
         0.02305542, -0.00747857],
       [-0.02819404, -0.04432267,  0.01159158, ..., -0.02953893,
         0.01612862, -0.0099255 ],
       [-0.0326709 , -0.0484228 ,  0.01606839, ..., -0.03584684,
         0.00761068, -0.00948259]], dtype=float32)
```

Word Vectors

Word Vectors : Context / Meaning + Relationships



Word Embeddings

- Word vectors can be obtained using the following techniques:
  - Training of word embedding representations from scratch
  - Pre-trained word embeddings:
    - word2vec
    - GloVe