NLP & Text Classification

NLP

Natural language processing (NLP) is a subfield of computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data.

Challenges in natural language processing frequently involve speech recognition, natural language understanding, and natural language generation.

(or anything to do with text…)

Wikipedia

https://en.wikipedia.org/wiki/Natural_language_processing

Language sucks…

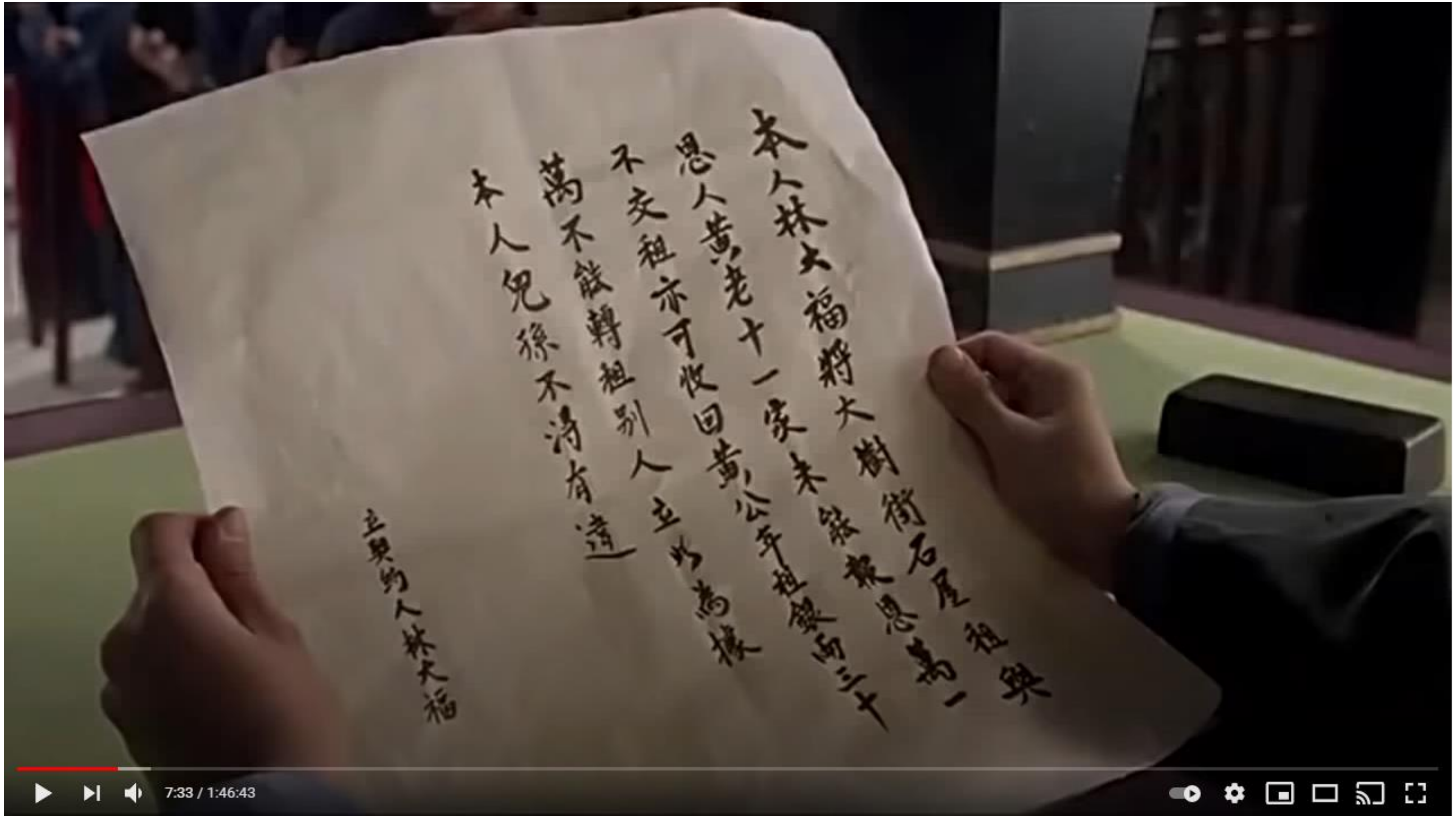
- 我頭有一點痛。
- 我有一點頭痛。

請問哪一句頭比較痛??

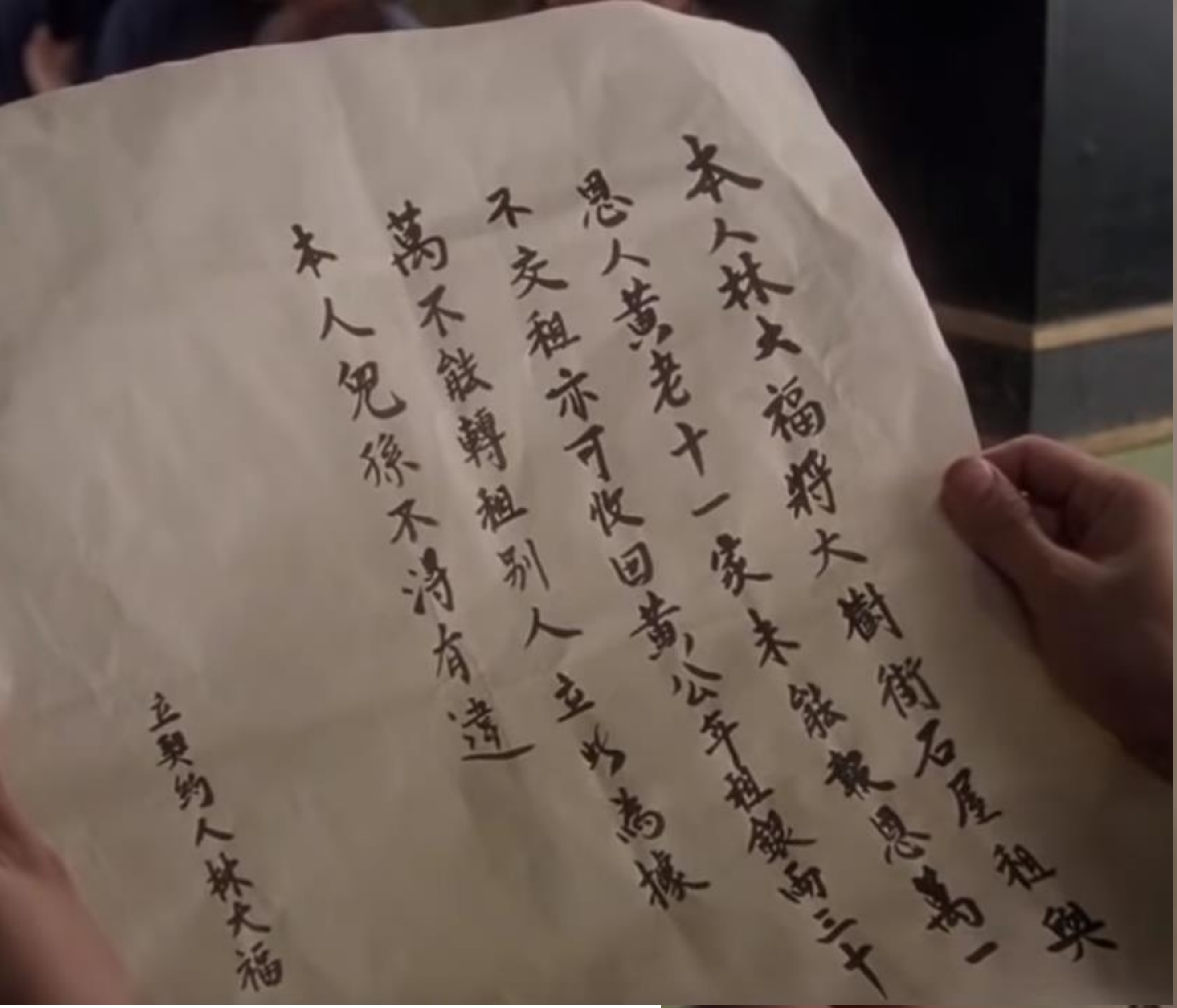
問題：文字組織鬆散

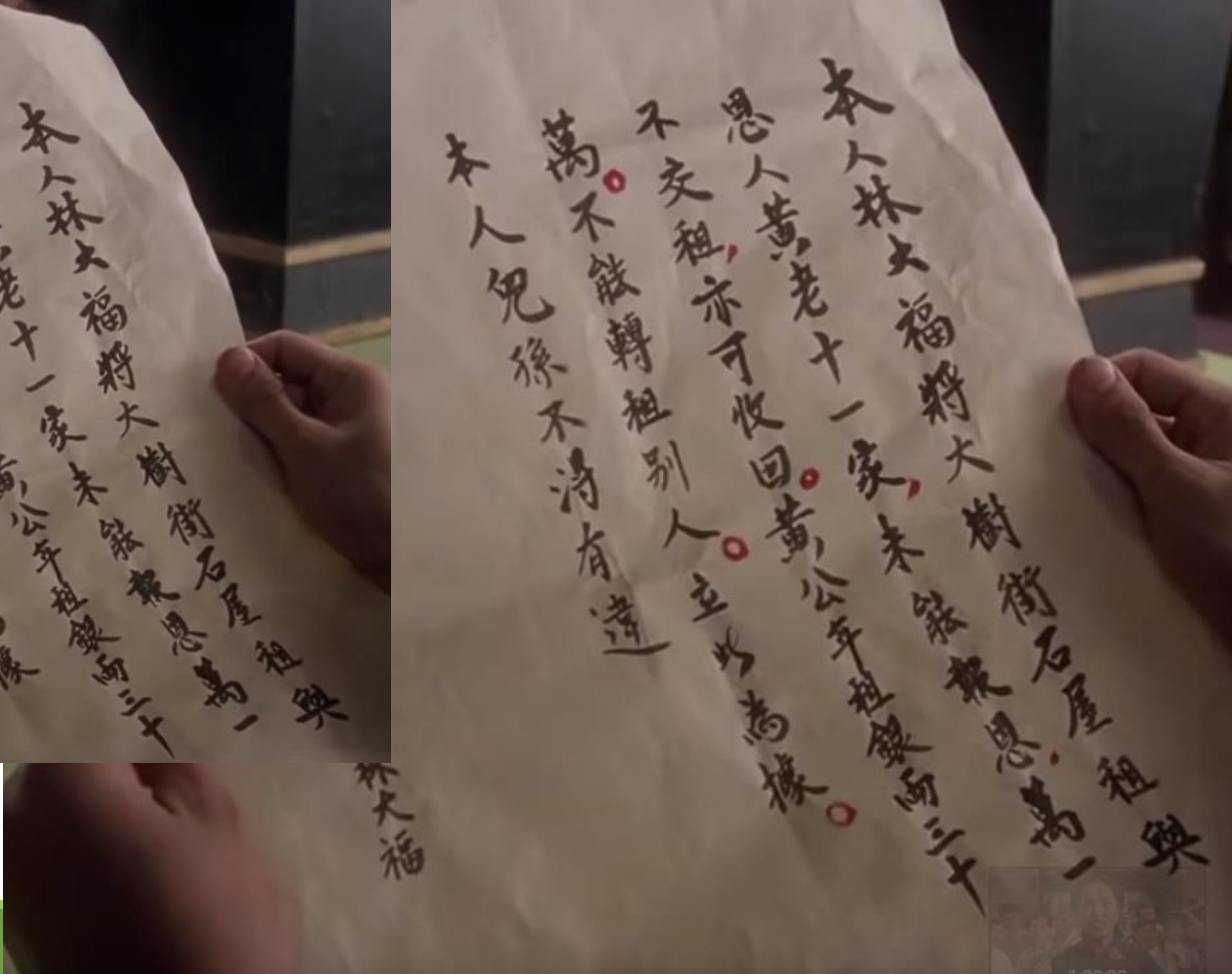
造句題目：難過

問題：文義模糊性



https://youtu.be/LXjusJ9ud78?t=448





問題：周星馳沒念書。

問題：斷句不清。

Languages vs Computers

"Languages express meaning by relating a sign form to a meaning, or its content. Sign forms must be something that can be perceived, for example, in sounds, images, or gestures, and then related to a specific meaning by **social convention**."

"Thus, languages must have a **vocabulary** of signs related to specific meaning."

A vocabulary is a set of familiar words within a person's language.

– Wikipedia

Computers only understand 0101010101…

Computers are good at numbers and algorithms…

But computers…

- do not inherently have social convention…
- do not understand the semantics like we humans do…

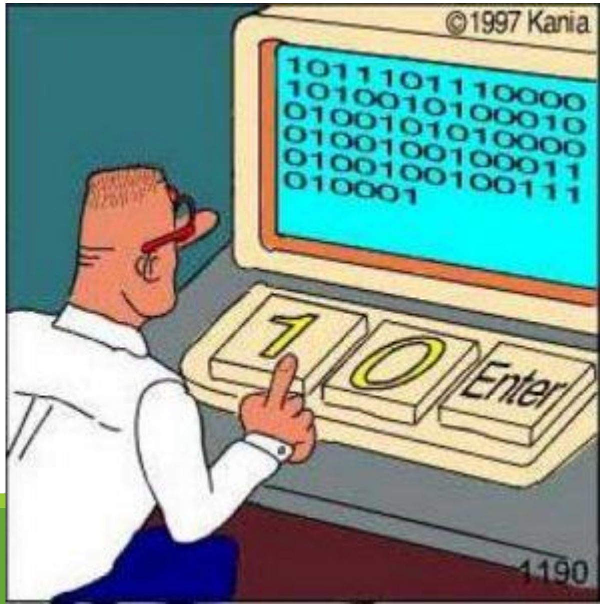


Image source: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/

Making Computers Understand ~~Languages~~ ENGLISH

London is the capital and most populous city of England and the United Kingdom. Standing on the River Thames in the south east of the island of Great Britain, London has been a major settlement for two millennia. It was founded by the Romans, who named it Londinium. London's ancient core, the City of London, largely retains its 1.12-square-mile (2.9 km2) medieval boundaries.

ON FIRE ??

Image source: https://medium.com/@ageitgey/natural-language-processing-is-fun-9a0bff37854e | https://gifer.com/en/PXzp | https://giphy.com/gifs/problems-nrXif9YExO9EI

Divide & Conquer!!

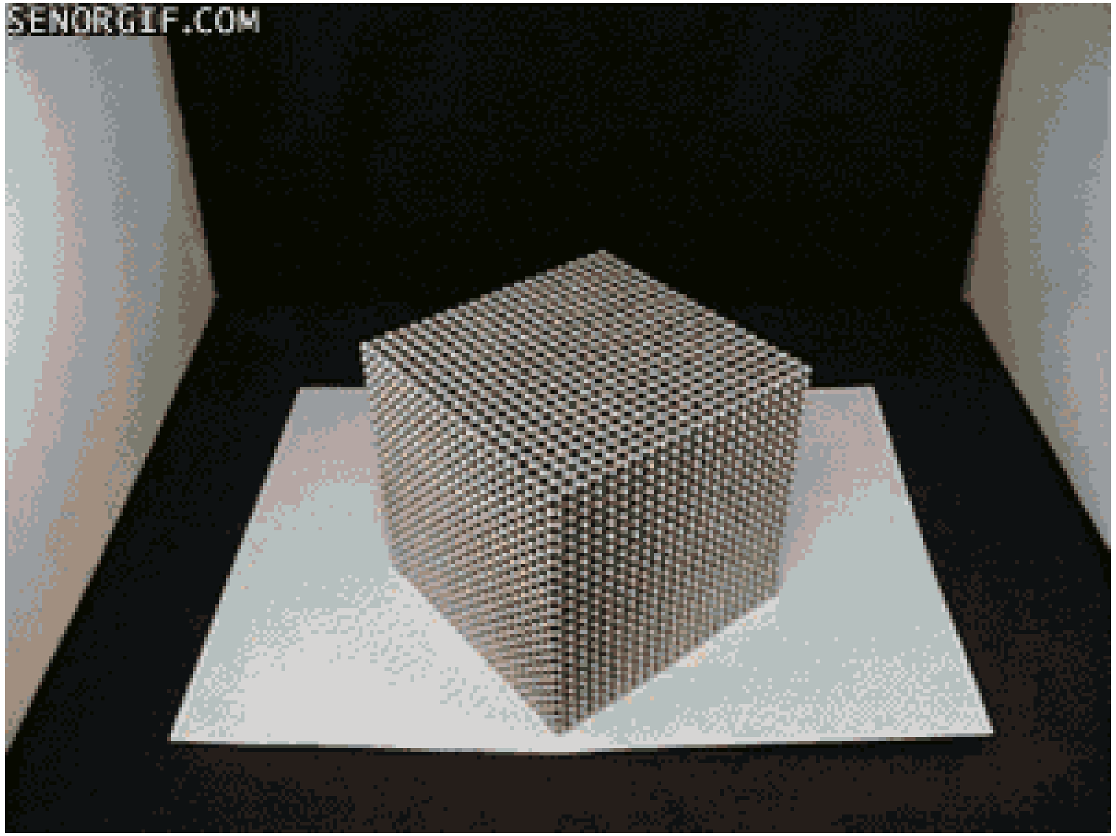


Image source: https://giphy.com/gifs/divide-conquer-GLkQ1upVWKdeo

Different preprocessing steps

1. Collect data
2. Sentence segmentation
3. Word tokenization (sometimes along with lowercasing)
4. Parts of Speech (POS) tagging (optional)
5. Lemmatization (or stemming)
6. Stop words removal
7. Dependency parsing
8. Named entity recognition
9. Coreference resolution

https://cloud.google.com/natural-language/
http://corenlp.run/

*** Note: steps are done according to your applications and not necessarily in the above order.

Collect Data

UCI Machine Learning Repository

Kaggle datasets

Yahoo Labs

Open Data on AWS

List of data repository on Kdnuggets

Some open datasets

Collect your own data

- Scrapy, BeautifulSoup, Selenium

Text Processing (Rule of Thumb)

Remove all irrelevant characters such as any non alphanumeric characters

Tokenize your text by separating it into individual words

Remove words that are not relevant, such as “@” twitter mentions or URLs

Convert all characters to lowercase, in order to treat words such as “hello”, “Hello”, and “HELLO” the same

Consider combining misspelled or alternately spelled words to a single representation (e.g. “cool”/”kewl”/”cooool”)

Consider lemmatization (reduce words such as “am”, “are”, and “is” to a common form such as “be”)

Sentence Segmentation

This is the Hotel to stay! If you are going to spend a few days in Beijing and looking for a hotel with great breakfast, excellent service, spacious room, easy to going around and with a very reasonable price, this is the best hotel to stay! We are a 6 people family group from Norway (including 2 kids aged at 8 and 10, 2 Grand parents aged over 60), by carefully reading the reviews on Beijing hotels from different travel agents, we finally picked up Central Plaza Holiday Inn (which is #1 on trip adviser list) as our one week stay in Beijing from 18th, sep. We made contact with Storm prior to our arrival via his email to help us book the hotel, he quickly made us a reservation with a very reasonable price, he also asked us if we need to be picked up in the airport and our arriving time, we did not use the pick up since we need meet our relatives first in the airport. When we arrived in the hotel, Storm surprised us by being there waiting for us (he actually worked additional hours just to wait for us and make sure everything is OK). He quickly asked our plan and also suggested a local Chinese restaurant which turns out to be our favourite place to eat during the week stay (Qing Nian Can Ting), 5 minutes walk from the hotel, and right besides a small supermarket.

This is the Hotel to stay!

If you are going to spend a few days in Beijing and looking for a hotel with great breakfast, excellent service, spacious room, easy to going around and with a very reasonable price, this is the best hotel to stay!

We are a 6 people family group from Norway (including 2 kids aged at 8 and 10, 2 Grand parents aged over 60), by carefully reading the reviews on Beijing hotels from different travel agents, we finally picked up Central Plaza Holiday Inn (which is #1 on trip adviser list) as our one week stay in Beijing from 18th, sep.

We made contact with Storm prior to our arrival via his email to help us book the hotel, he quickly made us a reservation with a very reasonable price, he also asked us if we need to be picked up in the airport and our arriving time, we did not use the pick up since we need meet our relatives first in the airport.

When we arrived in the hotel, Storm surprised us by being there waiting for us (he actually worked additional hours just to wait for us and make sure everything is OK).

He quickly asked our plan and also suggested a local Chinese restaurant which turns out to be our favourite place to eat during the week stay (Qing Nian Can Ting), 5 minutes walk from the hotel, and right besides a small supermarket.

This is the Hotel to stay!

If you are going to spend a few days in Beijing and looking for a hotel with great breakfast, excellent service, spacious room, easy to going around and with a very reasonable price, this is the best hotel to stay!

We are a 6 people family group from Norway (including 2 kids aged at 8 and 10, 2 Grand parents aged over 60), by carefully reading the reviews on Beijing hotels from different travel agents, we finally picked up Central Plaza Holiday Inn (which is #1 on trip adviser list) as our one week stay in Beijing from 18th, sep.

We made contact with Storm prior to our arrival via his email to help us book the hotel, he quickly made us a reservation with a very reasonable price, he also asked us if we need to be picked up in the airport and our arriving time, we did not use the pick up since we need meet our relatives first in the airport.

When we arrived in the hotel, Storm surprised us by being there waiting for us (he actually worked additional hours just to wait for us and make sure everything is OK).

He quickly asked our plan and also suggested a local Chinese restaurant which turns out to be our favourite place to eat during the week stay (Qing Nian Can Ting), 5 minutes walk from the hotel, and right besides a small supermarket.

Stemming vs Lemmatization

Background…

Playing → Play
Plays → Play
Played → Play

Common root form 'play'

am, are, is → be

Car cars, car's, cars' → car

Using above mapping a sentence could be normalized as follows:

the boy's cars are different colors → the boy car be differ color

DataCamp

*"In grammar, inflection is the modification of a word to express different grammatical categories such as tense, case, voice, aspect, person, number, gender, and mood. An inflection expresses one or more grammatical categories with a **prefix, suffix or infix**, or another internal modification such as a vowel change" [Wikipedia]*

Prefix, suffix, infix?? [Link 1](), [Link 2]()

Source: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

Stemming vs Lemmatization

Stemming and Lemmatization are **Text Normalization** (or sometimes called **Word Normalization**) techniques in the field of **Natural Language Processing** that are used to prepare text, words, and documents for further processing. Stemming and Lemmatization have been studied, and algorithms have been developed in Computer Science since the 1960's.

Stemming and Lemmatization helps us to achieve the root forms (sometimes called synonyms in search context) of inflected (derived) words. *Stemming is different to Lemmatization in the approach it uses to produce root forms of words and the word produced.*

Stemming and Lemmatization are widely used in **tagging systems, indexing, SEOs, Web search results, and information retrieval**. For example, searching for *fish* on Google will also result in *fishes*, *fishing* as *fish* is the stem of both words.

Source: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

Stemming

Stem (root) is the part of the word to which you add inflectional (changing/deriving) affixes such as (-ed,-ize, -s,-de,mis). So stemming a word or sentence may result in words that are not actual words. Stems are created by removing the suffixes or prefixes used with a word.

"Stemming is the process of reducing inflection in words to their root forms such as mapping a group of words to the same stem even if the stem itself is not a valid word in the Language."

Source: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

Stemming

Words having the same stem will have a similar meaning. For example,

CONNECT
CONNECTIONS------> CONNECT
CONNECTED------> CONNECT
CONNECTING------> CONNECT
CONNECTION------> CONNECT

http://textanalysisonline.com/

https://9ol.es/porter_js_demo.html

Source: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

Lemmatization

Lemmatization, unlike Stemming, reduces the inflected words properly ensuring that the root word belongs to the language. In Lemmatization root word is called ***Lemma****. A lemma (plural lemmas or lemmata) is the canonical form, dictionary form, or citation form of a set of words.*

For example, *runs, running, ran* are all forms of the word *run*, therefore *run* is the lemma of all these words. Because lemmatization returns an actual word of the language, it is used where it is necessary to get valid words.

Source: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

Lemmatization

sentence =

"He was running and eating at same time. He has bad habit of swimming after playing long hours in the Sun.“

Word	Lemma	With context
He	He	He
was	wa	be
running	running	run
and	and	and
eating	eating	eat
at	at	at
same	same	same
time	time	time
He	He	He
has	ha	have
bad	bad	bad
habit	habit	habit
of	of	of
swimming	swimming	swim
after	after	after
playing	playing	play
long	long	long
hours	hour	hours
in	in	in
the	the	the
Sun	Sun	Sun

Source: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

Should I use stemming or lemmatization?

- Stemming and Lemmatization both generate the root form of the inflected words. The difference is that stem might not be an actual word whereas, lemma is an actual language word.
- Stemming follows an algorithm with steps to perform on the words which makes it faster. Whereas, in lemmatization, you used WordNet corpus and a corpus for stop words as well to produce lemma which makes it slower than stemming. You also had to define a parts-of-speech to obtain the correct lemma.
- If speed is focused then stemming should be used since lemmatizers scan a corpus which consumed time and processing. It depends on the application you are working on that decides if stemmers should be used or lemmatizers. If you are building a language application in which language is important you should use lemmatization as it uses a corpus to match root forms.

Source: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

Stemming & Lemmatization

Stemming and Lemmatization are itself form of NLP and widely used in Text mining. Text mining tasks include **text categorization (aka text classification)**, **text clustering**, **concept/entity extraction**, **production of granular taxonomies**, **sentiment analysis**, **document summarization**, and **entity relation modeling (i.e., learning relations between named entities)**.

Source: https://www.datacamp.com/community/tutorials/stemming-lemmatization-python

Text Classification

What is Text Classification

TC is commonly referred to as “the task of classifying natural language documents into a pre-defined set of semantic categories”.

For example: Entertainment, Health, Business, Technology etc.

Motivation of Automatic TC

Categorised data are easier for users to browse

Organisational view of data provides more effective retrieval

Efficient search is not enough

http://www.bbc.co.uk/directory/

Messenger Express | BBC - Directory | BBC NEWS | Business | UK factory inp... | BBC NEWS | Business | Average hous...

bbc.co.uk — Home | TV | Radio | Talk | Where I Live | A-Z Index | Search

Accessibility help
Text Only

directory

Monday 12th February 2007

Business & Money

- Benefits & Tax Credits
- Borrowing & Debt
- Companies
- Economy
- Life At Work
- Market Data
- Mortgages & Housing
- News
- Pensions
- Programmes
- Running A Business
- Savings & Investments
- Tax & Inheritance
- Your Money

CBBC

- Art
- Backstage
- Cartoons
- CBBC Search
- Games
- Message Board
- Music
- Sport
- Star Chat
- What's On
- Wild
- Your Life

CBeebies

- Birthdays
- eCards
- Fimbles
- Fun & Games
- Grownups
- Colour & Make
- Message Board
- My CBeebies
- Prizes
- Sing a Song
- Story Circle
- Teletubbies
- Tweenies
- Your Gallery

Popular Links

- News
- Popular and New on bbc.co.uk
- Sport
- Weather
- What's On TV
- What's On Radio

Search the A-Z index:
A-Z Index

Done

Motivation of Automatic TC

Manual text classification is time-consuming and expensive

- MEDLINE (National Library of Medicine) indexed over 600k citations in 2006 using MEdical Subject Headings (23,000 categories)
- Yahoo! Directories – over 500k categories

Preprocessing

Fatal drug mix killed US R&B star

Grammy-nominated R&B star Gerald Levert was killed by an accidental mixture of over-the-counter and prescription drugs according to a US coroner.

The singer, who died last November, had pain killers, anxiety medication and allergy drugs in his bloodstream, said Cleveland coroner Kevin Chartrand.

The official cause of death was acute intoxication, and the death was ruled to be accidental, he said.

Levert found fame in R&B trio LeVert, and had a UK top 10 hit with Casanova.

He also recorded as a solo artist, and worked with soul legends such as Anita Baker, Barry White and Patti LaBelle.

--- BBC Sunday, 11 February 2007, 13:03 GMT

Category: Music? Health? Entertainment? R&B? USA? Medicine? UK?

How Automatic TC is done: *Knowledge Engineering*

In the late 1980s

Knowledge Engineering

- Experts hand-craft classification rules
- Rules
  - Rule 1:(*R&B* **or** *star* **or** *soul*) **and** (*singer* **or** *artist*) → ***Music***
  - Rule 2:(*drug* **or** *prescription*) **and** *medication* → ***Medicine***
  - Rule 3:(*anxiety* **or** *pain* **or** *allergy*) **and** *acute* → ***Health***
  - Rule 4 :(*play* **or** *fame*) **and** *award* → ***Entertainment***
  - Rule …

How Automatic TC is done: *Knowledge Engineering*

Still inefficient and impractical when

- Number of categories is large
- Category definitions can change over time
- Personalised application where an expert/knowledge engineer is unavailable

Inconsistency issues as rule set gets larger

How Automatic TC is done: *Machine Learning*

Since 1990s

The learning algorithm is given a small set of manually classified documents (training documents/dataset)

- Documents to be classified are test documents/dataset

Produces a classification rule automatically

A.k.a a supervised learning problem

But, how do we make the learning algorithm learn from the training documents?

How Automatic TC is done: *Machine Learning* - *Preprocessing*

Pre-processing

- Representing Text
  - Bag-of-words approach* – Term Frequency (TF)
  - Feature selection
    - Stopword removal
  - Feature construction
    - Stemming
    - Term weighting – DF, IDF

*bag-of-words approach may not be the best method for other languages

Preprocessing

Fatal drug mix killed US R&B star

Grammy-nominated R&B star Gerald Levert was killed by an accidental mixture of over-the-counter and prescription drugs according to a US coroner.

The singer, who died last November, had pain killers, anxiety medication and allergy drugs in his bloodstream, said Cleveland coroner Kevin Chartrand.

The official cause of death was acute intoxication, and the death was ruled to be accidental, he said.

Levert found fame in R&B trio LeVert, and had a UK top 10 hit with Casanova.

He also recorded as a solo artist, and worked with soul legends such as Anita Baker, Barry White and Patti LaBelle.

--- BBC Sunday, 11 February 2007, 13:03 GMT

Preprocessing

Fatal drug mix killed US R   B star

Grammy-nominated R   B star Gerald Levert was killed by an accidental mixture of over-the-counter and prescription drugs according to a US coroner

The singer who died last November had pain killers anxiety medication and allergy drugs in his bloodstream said Cleveland coroner Kevin Chartrand

The official cause of death was acute intoxication and the death was ruled to be accidental he said

Levert found fame in R B trio LeVert and had a UK top 10 hit with Casanova

He also recorded as a solo artist and worked with soul legends such as Anita Baker Barry White and Patti LaBelle

--- BBC Sunday, 11 February 2007, 13:03 GMT

Preprocessing

Fatal drug mix killed US R B star

Grammy-nominated R B star Gerald Levert killed accidental mixture over-the-counter prescription drugs according US coroner

singer died last November pain killers anxiety medication allergy drugs bloodstream Cleveland coroner Kevin Chartrand.

official cause death acute intoxication death ruled accidental

Levert found fame R B trio LeVert UK top 10 hit Casanova

recorded solo artist worked soul legends Anita Baker Barry White Patti LaBelle

--- BBC Sunday, 11 February 2007, 13:03 GMT

Preprocessing

Fatal drug mix kill US R B star

Grammy-nominat R B star Gerald Levert kill accident mix over-the-count prescri drug according US coron

sing die last Novemb pain kill anxiety medic allergy drug bloodstream Cleveland coron Kevin Chartrand.

offic caus death acut intoxic death rul accident

Levert found fame R B trio LeVert UK top 10 hit Casanova

record solo art work soul legend Anita Bak Barry Whit Patti LaBell

--- BBC Sunday, 11 February 2007, 13:03 GMT

Preprocessing

Fatal drug mix kill US R B star

Grammy-nominat R B star Gerald Levert kill accident mix over-the-count prescri drug according US coron

sing die last Novemb pain kill anxiety medic allergy drug bloodstream Cleveland coron Kevin Chartrand.

offic caus death acut intoxic death rul accident

Levert found fame R B trio LeVert UK top 10 hit Casanova

record solo art work soul legend Anita Bak Barry Whit Patti LaBell

--- BBC Sunday, 11 February 2007, 13:03 GMT

drug	play	album	award	accident	fame	…	kill	music
3	0	0	0	2	1		3	0

Machine Learning & Text

Word as feature/dimension.

Texts are represented by "vectors".

I love you.

I really like you.

I have feelings for you.

I really hate you.

feelings	for	hate	have	I	like	love	really	you
0	0	0	0	1	0	1	0	1
0	0	0	0	1	1	0	1	1
1	1	0	1	1	0	0	0	1
0	0	1	0	1	0	0	0	1

https://colab.research.google.com/drive/1TGaZosIeIGxtpLv-6OYS63ACl_qKiuJd

https://colab.research.google.com/drive/1f7FYwe-HZNnXD7DPx1bLjpGIaBOde5nx

Bag-of-Word, TF-IDF

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	0.18	**0.48**	0.18							
Doc 2	0.18		0.18	**0.48**	0.18	0.18				
Doc 3					0.18	0.18	**0.48**	**0.95**	**0.48**	**0.48**

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j

df_i = number of documents containing i

N = total number of documents

TF, TF-IDF

Term Frequency

Inverse Document Frequency

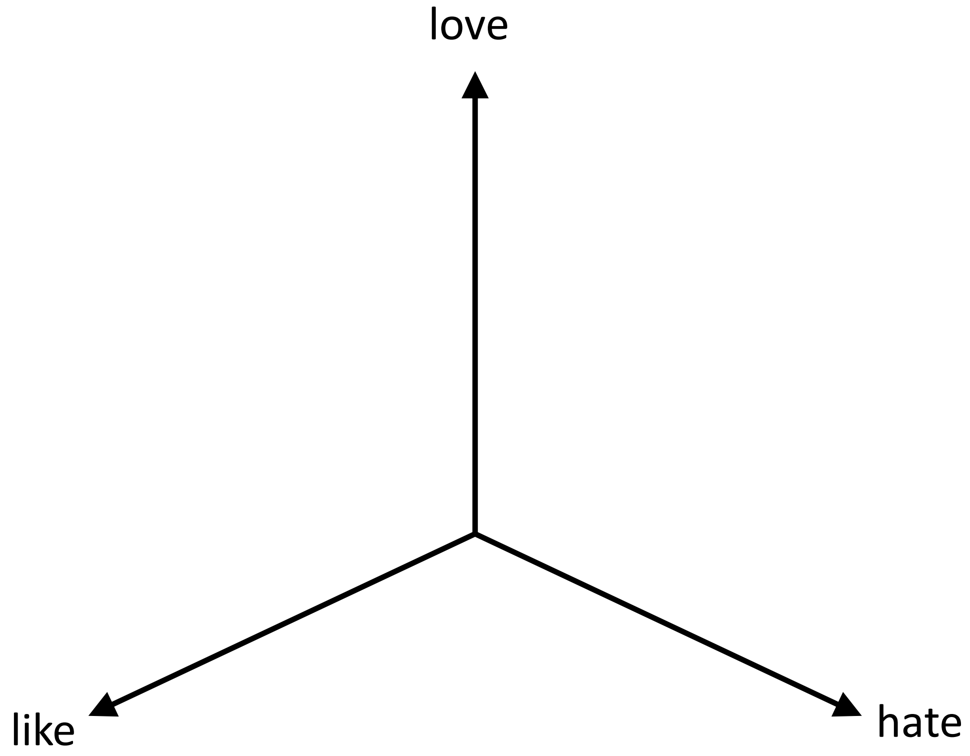
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

http://www.tfidf.com

https://medium.freecodecamp.org/how-to-process-textual-data-using-tf-idf-in-python-cd2bbc0a94a3

https://www.kdnuggets.com/2018/08/wtf-tf-idf.html

Machine Learning & Text



I don’t like him. I hate him.

I don’t hate him. I like him.

Bag-of-Word, TF-IDF

https://skymind.ai/wiki/bagofwords-tf-idf

http://datameetsmedia.com/bag-of-words-tf-idf-explained/

https://medium.com/deep-learning-turkey/text-processing-1-old-fashioned-methods-bag-of-words-and-tfxidf-b2340cc7ad4b

https://www.oreilly.com/library/view/feature-engineering-for/9781491953235/ch04.html

https://www.kaggle.com/reiinakano/basic-nlp-bag-of-words-tf-idf-word2vec-lstm

General Text Preprocessing

https://www.kdnuggets.com/2018/08/practitioners-guide-processing-understanding-text-2.html

https://www.kdnuggets.com/2018/03/text-data-preprocessing-walkthrough-python.html

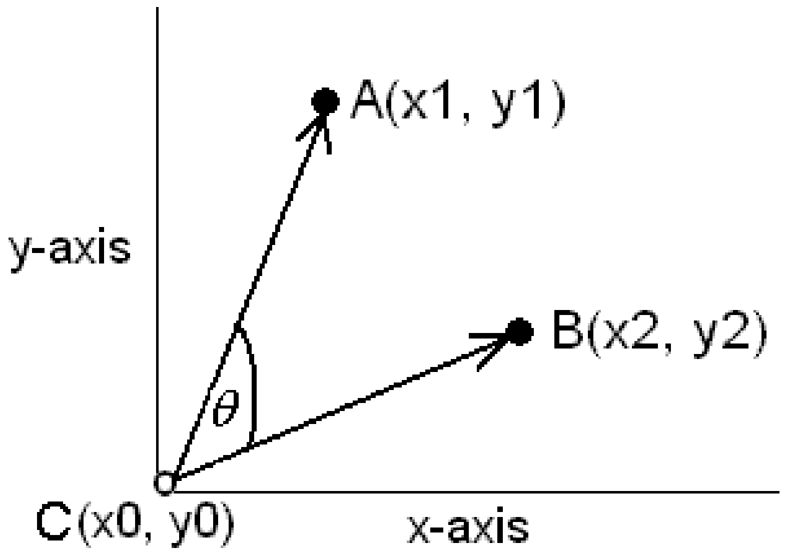
How Automatic TC is done: *Machine Learning - kNN*

k-Nearest Neighbour (kNN)

- Documents located close to each other are more likely to belong to the same class
- k is a pre-defined parameter, which determines how many "neighbouring" training documents to be considered when classifying a test document
- k is an integer = 1, 3 ,5, 7, 10…

Cosine Similarity is commonly used to determine the closeness of two documents

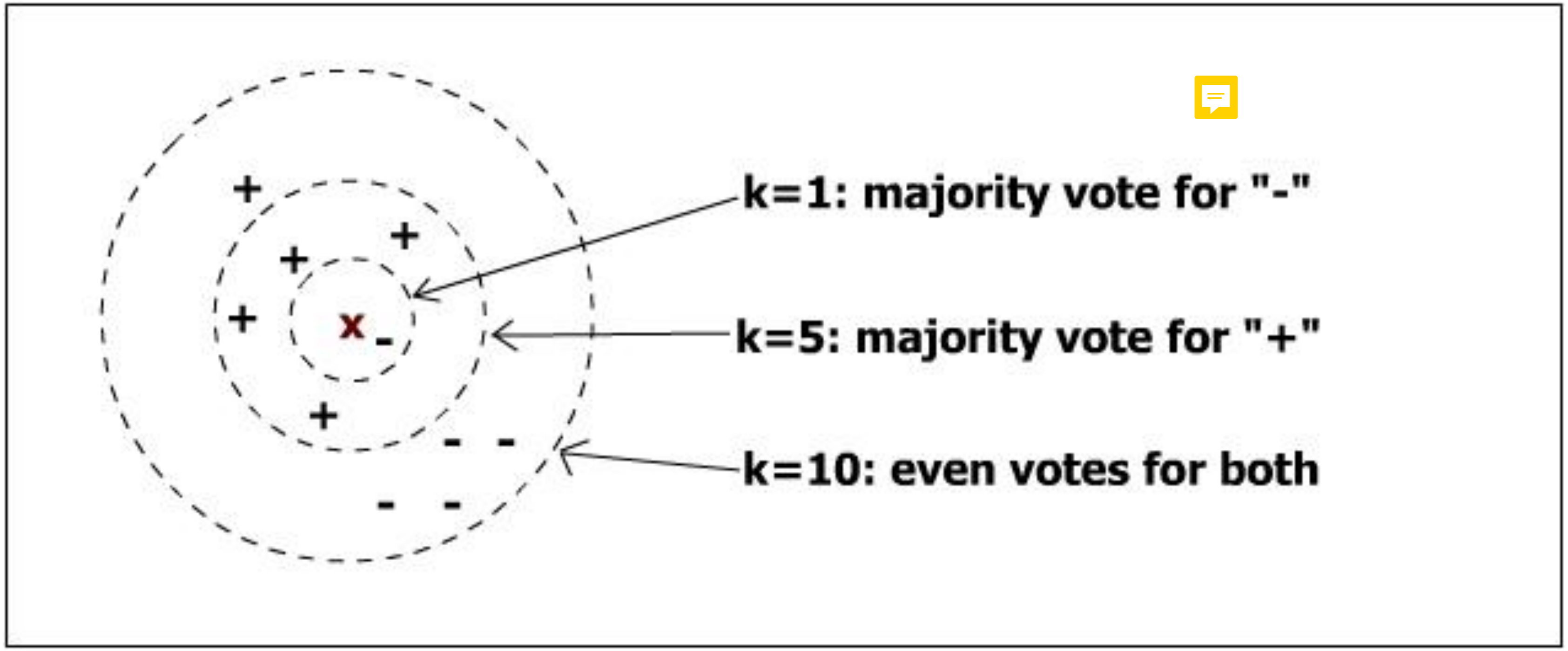
How Automatic TC is done: *Machine Learning - kNN*



$$\text{Sim}(A, B) = \text{cosine } \theta = \frac{A \bullet B}{|A||B|} = \frac{x1 * x2 + y1 * y2}{(x1^2 + y1^2)^{1/2} \ (x2^2 + y2^2)^{1/2}}$$

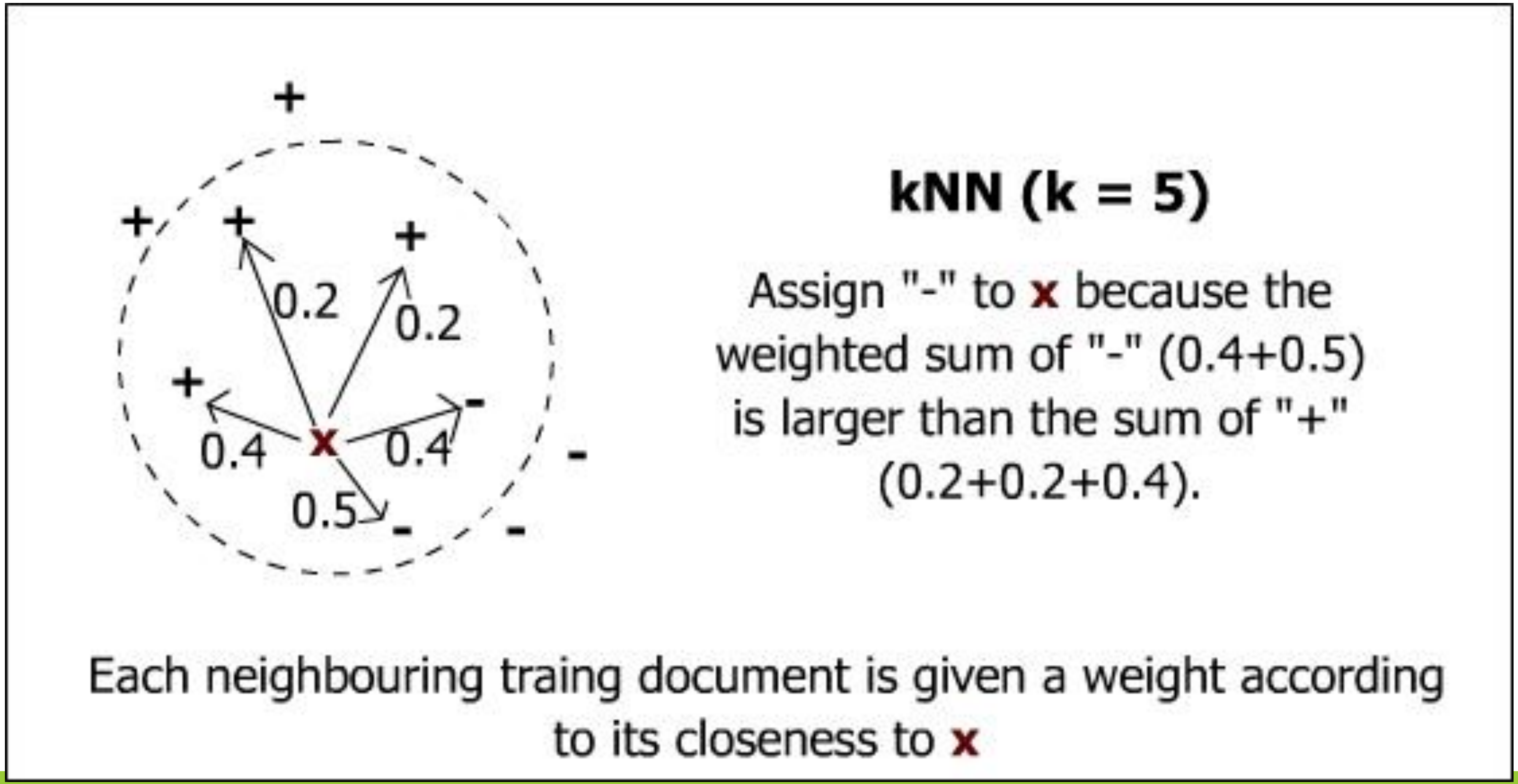
How Automatic TC is done: *Machine Learning - kNN*

Majority voting scheme



How Automatic TC is done: *Machine Learning - kNN*

Weighted-sum voting scheme



How Automatic TC is done: *Machine Learning - kNN*

The score for a category is the sum of the similarity scores between the point to be classified and all of its k-neighbours that belong to the given category.

To restate:

$$score(c \mid x) = \sum_{d \in kNN of\ x} sim(x,d)\, I(d,c)$$

where *x* is the new point; *c* is a class (e.g. black or white);
d is a classified point among the k-nearest neighbours of *x*;
sim(*x*,*d*) is the similarity between *x* and *d*;
I(*d*,*c*) = 1 if point d belongs to class *c*;
I(*d*,*c*) = 0 otherwise.

Exercise

Imagine a language that is made up with five English letters, A, B, C, D and E with **B, D and E being stopwords**. The kNN system has been "trained" with 3 training documents, which belong to TWO different categories (see below) and the task is to classify a new document (test document) into one of the two categories using the process of automatic text classification with kNN (k=1).

Preprocessed Training Documents:

Category	Doc ID	Doc Text	Doc Vector
1	D1	A C C A A A	(4, 2)
2	D2	C C C C A	(1, 4)
2	D3	A C C C A	(2, 3)

Unpreprocessed Test Document:

Category	Doc ID	Doc Text	Doc Vector
?	D4	A C B B A A E A D	?

How we know it works

Given n test documents and m category in consideration, a classifier makes $n \times m$ binary decisions. A two-by-two contingency table can be computed for each category

	truly YES	truly NO
classifier YES	True Positive (TP)	False Positive (FP)
classifier NO	False Negative (FN)	True Negative (TN)

How we know it works

Performance measures

- Precision (*p*)
- Recall (*r*)
- F_1-measure
- Accuracy

How we know it works

Precision = TP/(TP+FP) where TP + FP > 0 (otherwise undefined).

- Of the times we predicted it was "in class", how often are we correct?

Recall = TP/(TP+FN) where TP + FN > 0 (o.w. undefined).

- Did we find all of those that belonged in the class?

How we know it works

F_1-measure = $2(p \cdot r)/(p + r)$

- The weighted harmonic mean of precision and recall
- Single performance measure to compare different learning algorithms

$$\text{Accuracy} = \frac{\text{No. TP for all categories}}{\text{No. all test documents}}$$

TC Tutorials with Python

http://radimrehurek.com/data_science_python/

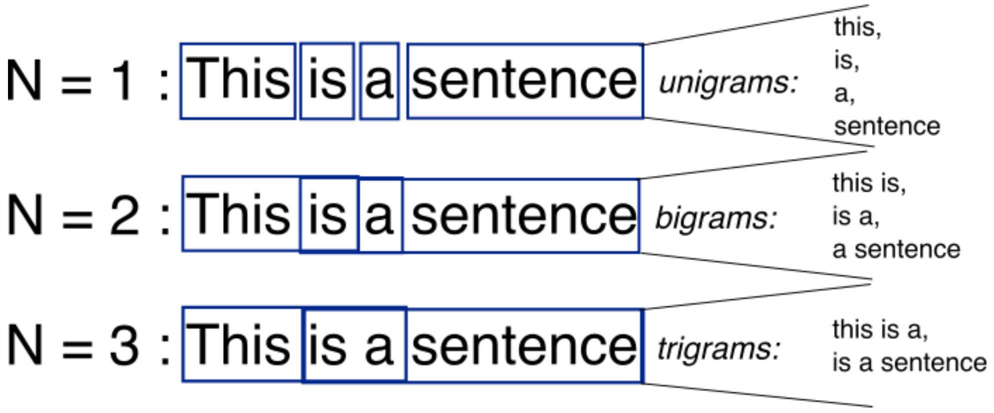
http://www.astroml.org/sklearn_tutorial/general_concepts.html

More NLP

N-Grams

An **n-gram** (also called multi-word unit or MWU) is a sequence of number of items (numbers, digits, words, letter etc.). In the context of text corpora, n-grams will typically refer to sequences of words. A **unigram** is one word, a **bigram** is a sequence of two words, a **trigram** is a sequence of three words etc. The items inside an n-gram may not have any relation between them apart from the fact that they appear next to each other.

https://www.sketchengine.eu/user-guide/user-manual/n-grams/



N-Grams

https://books.google.com/ngrams

https://stackoverflow.com/questions/18193253/what-exactly-is-an-n-gram

https://www.ngrams.info/

Part-Of-Speech (POS)

In the English language, words can be considered as the smallest elements that have distinctive meanings. Based on their use and functions, words are categorized into several types or parts of speech. This article will offer definitions and examples for the 8 major parts of speech in English grammar: noun, pronoun, verb, adverb, adjective, conjunction, preposition, and interjection.

http://partofspeech.org/

Part-Of-Speech (POS)

https://medium.com/@gianpaul.r/tokenization-and-parts-of-speech-pos-tagging-in-pythons-nltk-library-2d30f70af13b

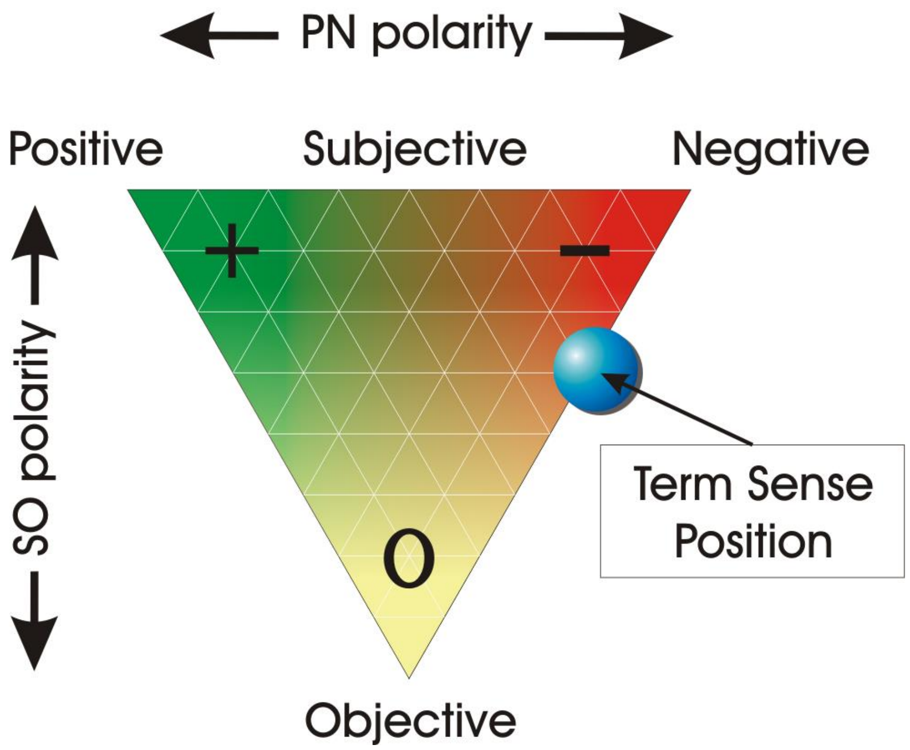
https://www.nltk.org/book/ch05.html

https://nlpforhackers.io/training-pos-tagger/

http://www.stokastik.in/building-a-pos-tagger-with-python-nltk-and-scikit-learn/

Sentiment… SentiwordNet (SWN)

SentiWordNet is a lexical resource in which each WordNet synset is associated to three numerical scores Obj(s), Pos(s) and Neg(s), describing how objective, positive, and negative the terms contained in the synset are.



http://ontotext.fbk.eu/sentiwn.html

https://swn.isti.cnr.it/

http://www.nltk.org/howto/sentiwordnet.html

Sentiment… SentiwordNet (SWN)

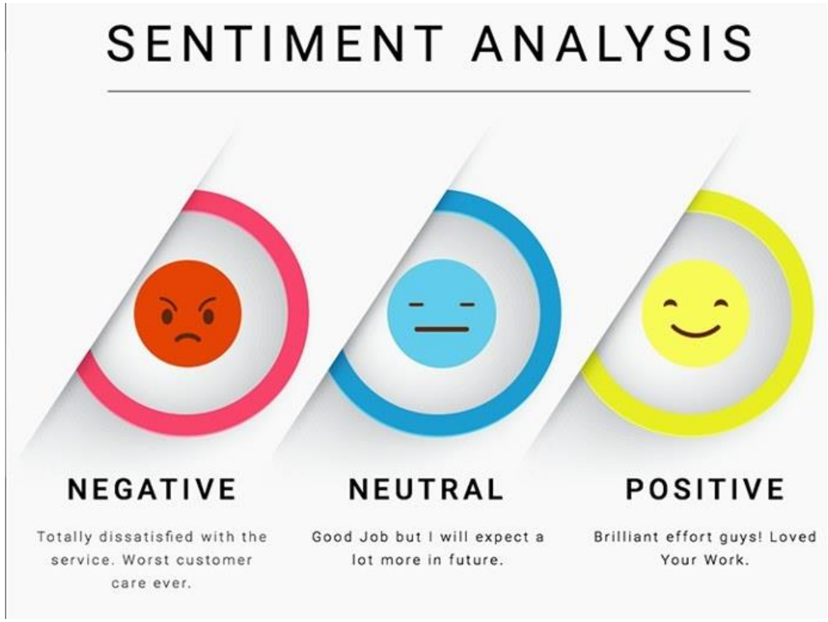
https://www.kaggle.com/nltkdata/sentiwordnet/version/1

https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17

https://www.kdnuggets.com/2018/03/5-things-sentiment-analysis-classification.html

https://monkeylearn.com/sentiment-analysis/

https://www.coursera.org/lecture/text-mining-analytics/5-6-how-to-do-sentiment-analysis-with-sentiwordnet-5RwtX



References & Resources

Text Wrangling & Pre-processing: A Practitioner’s Guide to NLP

- https://www.kdnuggets.com/2018/08/practitioners-guide-processing-understanding-text-2.html

Text Data Preprocessing: A Walkthrough in Python

- https://www.kdnuggets.com/2018/03/text-data-preprocessing-walkthrough-python.html

How to process textual data using TF-IDF in Python

- https://medium.freecodecamp.org/how-to-process-textual-data-using-tf-idf-in-python-cd2bbc0a94a3

Neural Coreference

- https://huggingface.co/coref/
- https://medium.com/huggingface/state-of-the-art-neural-coreference-resolution-for-chatbots-3302365dcf30
- https://medium.com/huggingface/how-to-train-a-neural-coreference-model-neuralcoref-2-7bb30c1abdfe

Word2Vec Implementation using Numpy

- https://github.com/datasciencekeke/word2vec_numpy
- https://towardsdatascience.com/an-implementation-guide-to-word2vec-using-numpy-and-google-sheets-13445eebd281
- https://towardsdatascience.com/word2vec-from-scratch-with-numpy-8786ddd49e72

Other Resources

Sebastiani, F. Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, Vol. 34, No. 1, 2002.

Joachims, T. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*, Kluwer Academic Publishers, 2002