ML7 – Natural Language Processing

Machine Learning – Tools and applications for policy – Lecture 8

Iman van Lelyveld – Michiel Nijhuis

DNB Data Science Hub

ML7 – Natural Language Processing

1. What are the main approaches in textual analysis?
2. Going beyond simple word counts
3. The arrival of Large Language Models (LLM)

Outline

ML7 – Natural Language Processing

Introduction

Examples

NLP Processing Layers
- Morphology
- Syntax
- Semantic

Wrap-up

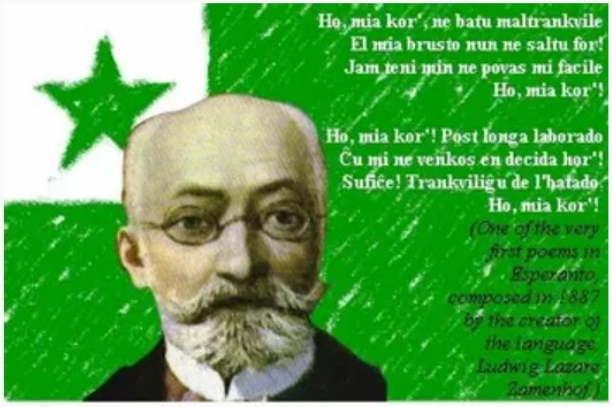
Introducing NLP

- What is Natural Language Processing (NLP)?
  - NLP is a field of Artificial Intelligence (AI)
  - NLP makes human language intelligible to machines.
  - Combines the power of linguistics and computer science to study the rules and structure of language
- What is NLP used for?
  - Rapid development of applications
  - e.g. automatically categorize email as Promotions, Social, Primary, or Spam
- With the arrival of Large Language Models such as ChatGPT the pace has picked up
- How does it work?

The vision

Gottfried Wilhelm Leibniz (1646 – 1716).

The vision



The creator of Esperanto, Ludwig Lazare Zamenhof, in front of the esperantist flag.

The vision

Source: Federation Starship Data Link

The vision

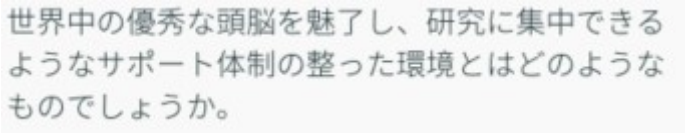
DataScience Hub

Common Examples of NLP

1. Email filters
2. Virtual assistants, voice assistants, or smart speakers
3. Online search engines
4. Predictive text and autocorrect
5. Monitor brand sentiment on social media
6. Sorting customer feedback and Chatbots
7. Natural language generation
8. Machine translation (MT)
9. Finding meaning and sentiment
10. Automatic summarization

Use Case: Machine translation (MT) I

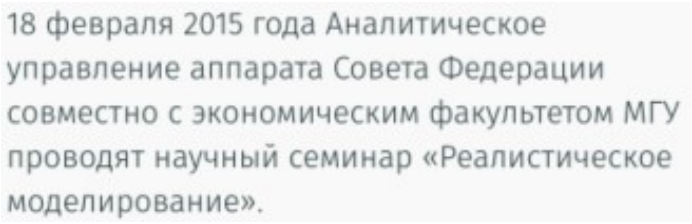
1. Interchange
   - Translation in one-to-one communication (telephone or written correspondence).
   - Internet: tweets, blog posts, forums
   - Human translation is out of the question (too slow)!
   - Any output (even if poor) is better than no output
2. Assimilation
   - Just to get a rough idea of the content
   - Output need not be perfect
   - But choice of words should reflect original meaning
   - Example Japanese–English translation, for assimilation:



   - *Attracts the brightest minds in the world, what What are the well-equipped environment support system, such as can concentrate on research*

Use Case: Machine translation (MT) II

3. Disemmination
   - Translation output to be distributed for human as-is without changes
   - End users will have high expectations! → output must be perfect
   - Hard – except for language pairs with huge amount of training data
   - Example Russian–English translation, suitable for dissemination:



   - *February 18, 2015 Analytical Department of the Federation Council in conjunction with the Faculty of Economics of Moscow State University conducted a scientific seminar "The realistic simulation."*

Use Case: Deciphering News and Speeches

- Deciphering the meaning of communications has a long history (e.g. Chappell et al. (1997) FOMC)
- Seminal contribution by Tetlock (2007)
- Surveys in Loughran and Mcdonald (2019) and Bholat et al. (2015)
- Various application in AlAjmi et al. (2023), Baker et al. (2016), Dim et al. (2021), Engle et al. (2020), Hassan et al. (2020), and Li et al. (2021) for:
  1. forecasting CDS
  2. changes in corporate culture
  3. economic uncertainty
  4. understanding the law
  5. and many, many more …

DataScience Hub

Would ML pick ”colour” words differently

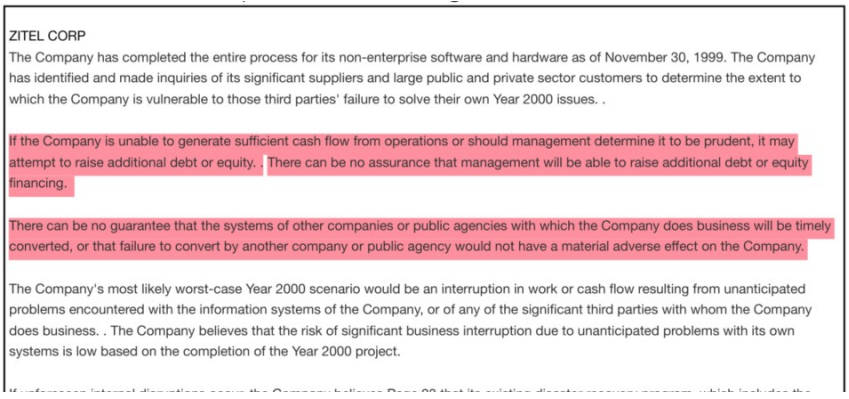
- García et al. (2023) take earnings calls and check which unigrams and bigrams are associated with stock price increases/decreases (ML dict.)
- They compare this to the performance of the widely used Loughran and McDonald dictionaries (LM dict.):
  - few new words but many LM words are not in ML list
  - bigrams with *leverage* is very positive in ML
  - holds up with other corpora (WSJ, 10-K)
- ”Man against machine”? Or ”machine against stock market”?

Top LM unigrams and ML scores. We consider the top 30 LM unigrams by frequency, separately for positive and negative words. For each of them, the table presents the total coverage (Cov., frequency over the whole earnings calls corpus measured in basis points), and the robust MNIR scores (positive and negative), namely the number of the 500 cross-validation samples for which that unigram is labelled as positive (negative) by the MNIR fit. Tokens coloured in blue (red) belong to the ML positive (negative) unigram dictionaries. (For interpretation of the references to colour in this table, the reader is referred to the web version of this article.)

Positive words				**Negative words**			
Token	Cov.	% Pos	% Neg	Token	Cov.	% Pos	% Neg
good^{+}	35.1	99.4	0.0	question	25.6	39.2	7.0
strong^{+}	26.0	100.0	0.0	questions	10.5	21.8	6.4
better^{+}	15.1	92.4	0.0	decline^{-}	8.0	0.0	99.8
opportunities	12.9	58.4	4.6	loss^{-}	6.8	0.0	99.0
able	12.1	63.2	2.2	negative^{-}	4.4	0.2	96.6
opportunity	11.9	68.0	3.8	difficult	3.7	0.0	78.4
positive	10.2	62.6	2.6	against	3.6	7.8	27.4
improvement^{+}	10.0	100.0	0.0	declined^{-}	3.5	0.2	91.4
progress	7.9	56.4	5.0	restructuring	3.2	30.8	30.4

DataScience Hub

Use Case: RiskFinder for automatic Summarization



ZITEL CORP

The Company has completed the entire process for its non-enterprise software and hardware as of November 30, 1999. The Company has identified and made inquiries of its significant suppliers and large public and private sector customers to determine the extent to which the Company is vulnerable to those third parties' failure to solve their own Year 2000 issues. .

If the Company is unable to generate sufficient cash flow from operations or should management determine it to be prudent, it may attempt to raise additional debt or equity. There can be no assurance that management will be able to raise additional debt or equity financing.

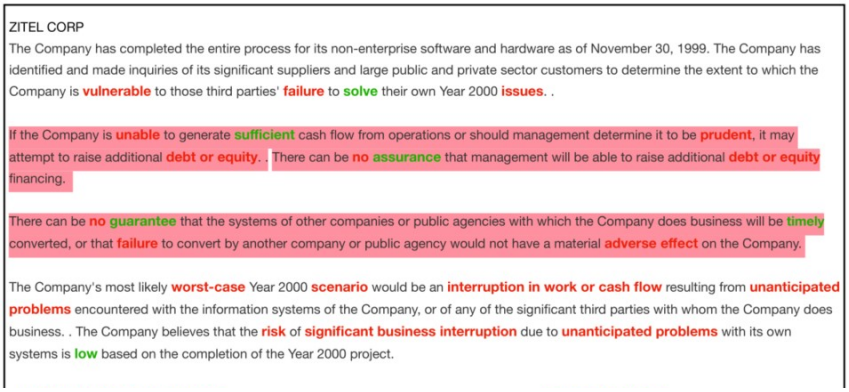
There can be no guarantee that the systems of other companies or public agencies with which the Company does business will be timely converted, or that failure to convert by another company or public agency would not have a material adverse effect on the Company.

The Company's most likely worst-case Year 2000 scenario would be an interruption in work or cash flow resulting from unanticipated problems encountered with the information systems of the Company, or of any of the significant third parties with whom the Company does business. . The Company believes that the risk of significant business interruption due to unanticipated problems with its own systems is low based on the completion of the Year 2000 project.

RiskFinder and Liu et al. (2018)

DataScience Hub

Use Case: RiskFinder for automatic Summarization



ZITEL CORP

The Company has completed the entire process for its non-enterprise software and hardware as of November 30, 1999. The Company has identified and made inquiries of its significant suppliers and large public and private sector customers to determine the extent to which the Company is **vulnerable** to those third parties' **failure** to **solve** their own Year 2000 **issues**. .

If the Company is **unable** to generate **sufficient** cash flow from operations or should management determine it to be **prudent**, it may attempt to raise additional **debt or equity**. . There can be **no assurance** that management will be able to raise additional **debt or equity** financing.

There can be **no guarantee** that the systems of other companies or public agencies with which the Company does business will be **timely** converted, or that **failure** to convert by another company or public agency would not have a material **adverse effect** on the Company.

The Company's most likely **worst-case** Year 2000 **scenario** would be an **interruption in work or cash flow** resulting from **unanticipated problems** encountered with the information systems of the Company, or of any of the significant third parties with whom the Company does business. . The Company believes that the **risk** of **significant business interruption** due to **unanticipated problems** with its own systems is **low** based on the completion of the Year 2000 project.

RiskFinder and Liu et al. (2018)

DataScience Hub

Use Case: Using sentiment for GDP forecasting

- DNB used NLP on millions of newspaper articles (36 years of history, Van Dijk and De Winter (2023))
- Result: a sentiment indicator that helps improve short run GDP forecasts
- Measures ecomic activities faster that existing measures from Central Bureau for Statistics

DNB-onderzoek: het FD voorspelt de economische trend in Nederland

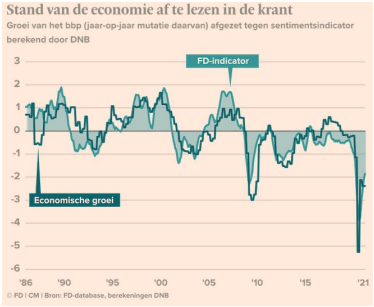
Mathijs Rotteveel Daan Ballegeer 26 mrt '21 21:00

Wie het FD leest, heeft voorkennis. Onderzoek van De Nederlandsche Bank leert namelijk dat het woordgebruik in de krant een voorspellende waarde heeft voor de economie.

DataScience Hub

Use Case: Using sentiment for GDP forecasting

- DNB used NLP on millions of newspaper articles (36 years of history, Van Dijk and De Winter (2023))
- Result: a sentiment indicator that helps improve short run GDP forecasts
- Measures ecomic activities faster that existing measures from Central Bureau for Statistics



DataScience Hub

Linguistic Layers in NLP

1. Morphology: word formation
   - Tokenising
   - Stemming
   - Lemmatising
2. Syntactic analysis: identifies the syntactic structure of a text and the dependency relationships between words, represented on a diagram called a parse tree (aka parsing or syntax analysis).
   - Part-of-Speech (POS)
   - Dependency
3. Semantics: aims to identify the meaning of language
   - Word Sense Disambiguation (WSD)
   - Sense tagging
4. Speech: phonemes
   - Distinct units of sound that distinguish one word from another: e.g. p, b, d, and t in the English words pad, pat, bad, and bat

Morphology: how words are formed

- Differs markedly between languages
- For English:

  Inflection: plant $\longrightarrow$ plants, planted, planting ...
  Derivation: plant $\longrightarrow$ plantation, implant ...
- For Indonesian:

  Inflection: sakit $\longrightarrow$ sakitnya;
  pergi $\longrightarrow$ pergilah
  Derivation: sakit $\longrightarrow$ pesakit, penyakit, sakitan...

Tokenising

- Tokenization: essential task in NLP used to break up a string of words into semantically useful units called tokens.
  - splits sentences, and word tokenization splits words within a sentence.

Tokenising

- Tokenization: essential task in NLP used to break up a string of words into semantically useful units called tokens.
  - splits sentences, and word tokenization splits words within a sentence.
- Just by space characters…?
  - Passers-by | didn't | go …

Tokenising

- Tokenization: essential task in NLP used to break up a string of words into semantically useful units called tokens.
  - splits sentences, and word tokenization splits words within a sentence.
- Just by space characters…?
  - Passers-by didn't go …
- Just by punctuation/word boundaries…?
  - Passers - by didn ' t go …

Tokenising

- Tokenization: essential task in NLP used to break up a string of words into semantically useful units called tokens.
  - splits sentences, and word tokenization splits words within a sentence.
- Just by space characters…?
  - [Passers-by] [didn't] [go] …
- Just by punctuation/word boundaries…?
  - [Passers] [-] [by] [didn] ['] [t] [go] …
- High-level tokenization works for more complex structures, like words that often go together (i.e. collocations: New York)

Tokenising

- Tokenization: essential task in NLP used to break up a string of words into semantically useful units called tokens.
  - splits sentences, and word tokenization splits words within a sentence.
- Just by space characters…?
  - [Passers-by] [didn't] [go] …
- Just by punctuation/word boundaries…?
  - [Passers] [-] [by] [didn] ['] [t] [go] …
- High-level tokenization works for more complex structures, like words that often go together (i.e. collocations: New York)
- Tokenizers need to consider natural language!
  - [Passers-by] [did] [n't] [go] …

Tokenising

- Tokenization: essential task in NLP used to break up a string of words into semantically useful units called tokens.
  - splits sentences, and word tokenization splits words within a sentence.
- Just by space characters. . . ?
  - [Passers-by] [didn't] [go] . . .
- Just by punctuation/word boundaries. . . ?
  - [Passers] [-] [by] [didn] ['] [t] [go] . . .
- High-level tokenization works for more complex structures, like words that often go together (i.e. collocations: New York)
- Tokenizers need to consider natural language!
  - [Passers-by] [did] [n't] [go] . . .
- Related: How to identify sentence boundaries?
  - "That's wonderful,' he said. 'Have your people call mine. Try to arrange something by 10 a.m. tomorrow."

Stemming

- Stem: reduced form (word stem, base or root form) or a word
- Need not be identical to the morphological root of the word!
- As long as related words map to the same stem
- Usually implemented by stripping prefix/suffix
- Example stemming:
  - producer → produc
  - produced → produc
  - producing → produc
  - carresses → carress
  - ponies → poni
  - caress → caress
  - cats → cat
- Can have phases/sequences of rules

Why Stemming?

- Information Retrieval – search for documents based on keywords
- Stem all words in documents and store as index
- Input keyword: producer $\rightarrow$ 'produc'
- Search documents whose indices contain 'produc'
- Results will include documents containing 'produce', 'produced', 'producer' ...

Lemmatising

- Lemma: base form of a word or term that is used as the formal dictionary entry for the term.
- Lemmatising can be seen as a special form of stemming
  - Stemming: outputs do not need to be real words
  - Lemmatising: outputs are genuine words used as headwords in dictionaries

(1) *Input:* *banks raised rates to fight inflation*
Lemmas: bank raise rates to fight inflation

Stemming vs Lemmatising

- Stemming is much faster than lemmatising
- But lemmatising is essential for many NLP tasks
- Would lemmatising be required for Chinese?

Segmentation

- Languages without word boundaries, e.g. Chinese, Thai, Japanese, German…

有 职称 的 和 尚未 有 职称 的
with position ones and not yet with position ones

有 职称 的 和尚 未有 职称 的
with position ones monks without position ones

Segmentation

- Languages without word boundaries, e.g. Chinese, Thai, Japanese, German…
- Essential for proper understanding!

有 职称 的 和 尚未 有 职称 的
with position ones and not yet with position ones

有 职称 的 和尚 未有 职称 的
with position ones monks without position ones

Segmentation

- Languages without word boundaries, e.g. Chinese, Thai, Japanese, German…
- Essential for proper understanding!
- Chinese example: 有职称的和尚未有职称的

有 职称 的 和 尚未 有 职称 的
with position ones and not yet with position ones

有 职称 的 和尚 未有 职称 的
with position ones monks without position ones

Outline

ML7 – Natural Language Processing

Introduction

Examples

NLP Processing Layers
- Morphology
- Syntax
- Semantic

Wrap-up

Syntax: words form phrases and sentences

- Grammatical rules and structures are known
- Syntactic processing: extract structure of phrase/sentences

1. Part-of-Speech (POS)
2. Parsing

Part-of-Speech (POS)

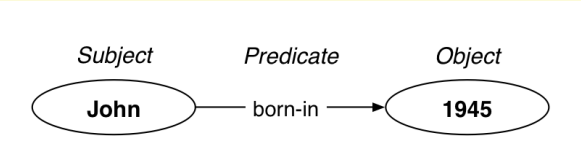
- Part-of-speech (PoS) tagging: add a speech category to each token within a text.
  - Common PoS tags are verb, adjective, noun, pronoun, etc
  - PoS tagging is useful for identifying relationships between words → understand the meaning of sentences

(2) *Input:* *banks raised rates to fight inflation*
POS-tags: NNS VBD NNS TO VB NN

Tag	Description
NN	Noun, singular or mass
NNS	Noun, plural
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
…	…

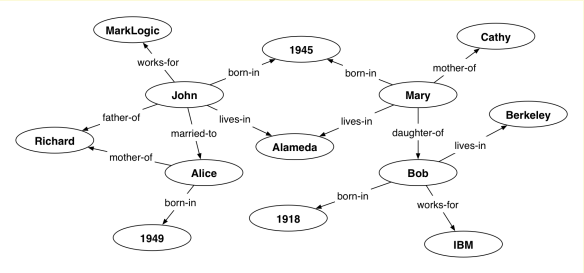
How to store annotations?

- The base of linked data is the Resource Description Format (RDF) more info
- Framework for describing resources on the world wide web
- Can be queried with SPARQL



How to store annotations?

- The base of linked data is the Resource Description Format (RDF) more info
- Framework for describing resources on the world wide web
- Can be queried with SPARQL



POS Tagset

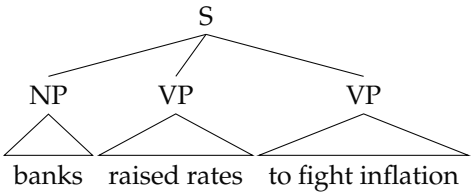
- Given an utterance, assign the most likely POS tag to each word token
- Current libraries quite stable now (for English): $\sim 96\%$ accuracy
- Different languages may have different sets of POS Tagsets
- English: Penn Treebank (PTB) tagset is widely adopted

Phrase and Sentence Structure

- Sentences/clauses are made up of phrases following grammar/syntax rules
- Some examples:
  - Noun phrase (NP): ‘a bright star’, ‘cats’, ‘stars and moons’
  - Verb phrase (VP): ‘ran’, ‘picks the ball up’
  - Clause/sentence (S): NP VP ‘a bright star picks the ball up’
- A syntactically correct sentence doesn’t have to makes sense!

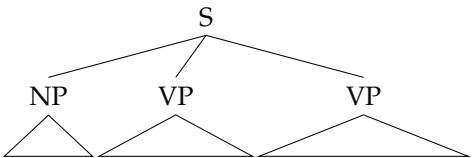
Shallow parsing or chunking

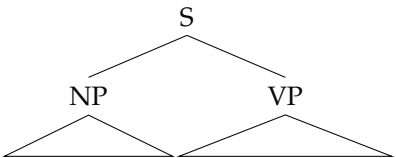
- Identify the noun phrases, verb phrases etc. but do not go into the internal structure



Shallow parsing or chunking

- Identify the noun phrases, verb phrases etc. but do not go into the internal structure



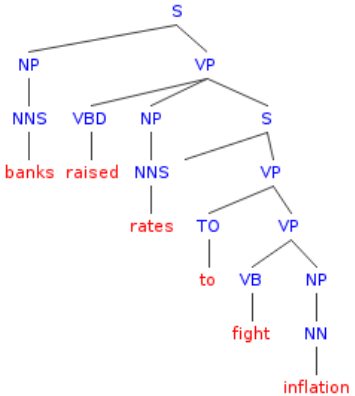


Deep parsing

- Fully building the clauses and relations in a sentence
- Syntactic parse tree:

‘Banks raised rates to fight inflation’

```
(S
  (NP (NNS banks))
  (VP (VBD raised)
    (NP (NNS rates))
    (S
      (VP (TO to)
        (VP (VB fight)
          (NP (NN inflation)))))))
```



Dependency parsing: find dependency relations

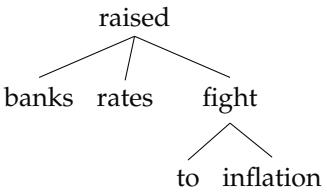
- Dependency Parsing: uncover the way the words in a sentence are connected. A dependency parser, therefore, analyzes how head words are related and modified by other words too understand the syntactic structure of a sentence:

```
nsubj(raised, banks)
root(ROOT, raised)
dobj(raised, rates)
aux(fight, to)
vmod(raised, fight)
dobj(fight, inflation)
```

'Banks raised rates to fight inflation'

- 'banks' is subject of 'raised'
- 'rates' is object of 'raised'
- …

Parsing is more difficult than POS-tagging



Outline

ML7 – Natural Language Processing

Introduction

Examples

NLP Processing Layers
- Morphology
- Syntax
- Semantic

Wrap-up

Semantic

- The meaning conveyed by the text
- Hard!
- How to represent ‘meaning’?
- Still an open question in artificial intelligence, cognitive science, psychology…
- Lots of ongoing research

Word Sense Disambiguation (WSD)

- Word Sense Disambiguation (WSD): Depending on their context, words can have different meanings (a.k.a. Sense-tagging). Take the word "book", for example:
  - You should read this book; it's a great novel!
  - You should book the flights as soon as possible.
  - You should close the books by the end of the year.
  - You should do everything by the book to avoid potential complications.
- Main techniques for WSD:
  1. knowledge-based (or dictionary approach): tries to infer meaning by observing the dictionary definitions of ambiguous terms within a text
  2. supervised approach: based on natural language processing algorithms that learn from training data
- Can be used for Synonym Expansion: Search for 'wizard' would also retrieve documents containing 'sorcerer', 'magician'

Word Sense Disambiguation (WSD)

- Stop words: Words that are ignored in NLP tasks (e.g. function words in a sense-tagging task). Filters out high-frequency words that add little value to a sentence, for example, which, to, at, for, is, etc.
- How to identify stop words?
  - Open-class words (content words): nouns, verbs, adjectives, adverbs
  - Closed-class words (function words): determiners, pronouns, conjunctions, infinitives...
  - Stop word are the residual

  ...so WSD needs POS-tagging and lemmatization first

WSD Example

Senses of bank.n in WordNet

1. sloping land (especially the slope beside a body of water)
2. a financial institution that accepts deposits and channels the money into lending activities
3. a long ridge or pile
4. …

(3)

Input:	*banks*	*raised*	*rates*	*to*	*fight*	*inflation*
Sense-tags:	bank.n.2	raise.v.13	rates.n.1		fight.v.1	inflation.n.1

Concept Tagging

- Label each sense in the input with a concept tag
  (Example below uses WordNet–SUMO mapping)

(4)

Input:	*banks*	*raised*	*rates*	*to*	*fight*	*inflation*
Sense-tags:	bank.n.2	raise.v.13	rates.n.1		fight.v.1	inflation.n.1
Concept tags:	CORPORATION	INCREASING	TAX		VIOLENTCONTEST	INCREASING

Extracting the right information

- Named Entity Recognition (NER): one of the most popular tasks in semantic analysis and involves extracting entities from within a text. Entities can be names, places, organizations, etc. Relationship extraction, another sub-task of NLP, goes one step further and finds relationships between two nouns. For example, in the phrase "Susan lives in Los Angeles," a person (Susan) is related to a place (Los Angeles) by the semantic category "lives in."
- Coreference resolution
  - 'The cat climbed onto the chair. It yawned and slept.'

Extracting the right information

- Named Entity Recognition (NER): one of the most popular tasks in semantic analysis and involves extracting entities from within a text. Entities can be names, places, organizations, etc. Relationship extraction, another sub-task of NLP, goes one step further and finds relationships between two nouns. For example, in the phrase "Susan lives in Los Angeles," a person (Susan) is related to a place (Los Angeles) by the semantic category "lives in."
- Coreference resolution
  - 'The cat climbed onto the chair. It yawned and slept.'
  - Is 'It' $\rightarrow$ 'the cat' or 'the chair'?

Extracting the right information

- Named Entity Recognition (NER): one of the most popular tasks in semantic analysis and involves extracting entities from within a text. Entities can be names, places, organizations, etc. Relationship extraction, another sub-task of NLP, goes one step further and finds relationships between two nouns. For example, in the phrase "Susan lives in Los Angeles," a person (Susan) is related to a place (Los Angeles) by the semantic category "lives in."
- Coreference resolution
  - 'The cat climbed onto the chair. It yawned and slept.'
  - Is 'It' $\rightarrow$ 'the cat' or 'the chair'?
  - 'cat' $\xrightarrow{\text{is-a}}$ ANIMAL $\xrightarrow{\text{is-a}}$ ANIMATE OBJECT

Extracting the right information

- Named Entity Recognition (NER): one of the most popular tasks in semantic analysis and involves extracting entities from within a text. Entities can be names, places, organizations, etc. Relationship extraction, another sub-task of NLP, goes one step further and finds relationships between two nouns. For example, in the phrase "Susan lives in Los Angeles," a person (Susan) is related to a place (Los Angeles) by the semantic category "lives in."
- Coreference resolution
  - 'The cat climbed onto the chair. It yawned and slept.'
  - Is 'It' $\rightarrow$ 'the cat' or 'the chair'?
  - 'cat' $\xrightarrow{\text{is-a}}$ ANIMAL $\xrightarrow{\text{is-a}}$ ANIMATE OBJECT
  - 'chair' $\xrightarrow{\text{is-a}}$ FURNITURE $\xrightarrow{\text{is-a}}$ INANIMATE OBJECT

Extracting the right information

- Named Entity Recognition (NER): one of the most popular tasks in semantic analysis and involves extracting entities from within a text. Entities can be names, places, organizations, etc. Relationship extraction, another sub-task of NLP, goes one step further and finds relationships between two nouns. For example, in the phrase "Susan lives in Los Angeles," a person (Susan) is related to a place (Los Angeles) by the semantic category "lives in."
- Coreference resolution
  - 'The cat climbed onto the chair. It yawned and slept.'
  - Is 'It' $\rightarrow$ 'the cat' or 'the chair'?
  - 'cat' $\xrightarrow{\text{is-a}}$ ANIMAL $\xrightarrow{\text{is-a}}$ ANIMATE OBJECT
  - 'chair' $\xrightarrow{\text{is-a}}$ FURNITURE $\xrightarrow{\text{is-a}}$ INANIMATE OBJECT
  - ANIMATE OBJECT $\xrightarrow{\text{capable-of}}$ 'yawn', 'sleep'

Extracting the right information

- Named Entity Recognition (NER): one of the most popular tasks in semantic analysis and involves extracting entities from within a text. Entities can be names, places, organizations, etc. Relationship extraction, another sub-task of NLP, goes one step further and finds relationships between two nouns. For example, in the phrase "Susan lives in Los Angeles," a person (Susan) is related to a place (Los Angeles) by the semantic category "lives in."
- Coreference resolution
  - 'The cat climbed onto the chair. It yawned and slept.'
  - Is 'It' $\rightarrow$ 'the cat' or 'the chair'?
  - 'cat' $\xrightarrow{\text{is-a}}$ ANIMAL $\xrightarrow{\text{is-a}}$ ANIMATE OBJECT
  - 'chair' $\xrightarrow{\text{is-a}}$ FURNITURE $\xrightarrow{\text{is-a}}$ INANIMATE OBJECT
  - ANIMATE OBJECT $\xrightarrow{\text{capable-of}}$ 'yawn', 'sleep'
  - $\therefore$ 'It' = 'the cat'

Extracting the sentiment

- Text Classification: aims to understand the meaning of unstructured text and organizing it into predefined categories (tags).
- One of the most popular text classification tasks is sentiment analysis, which aims to categorize unstructured data by sentiment.
  - Other classification tasks: intent detection, topic modeling, and language detection.

Extracting sentiment in practice: NLTK's VADER


```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentences = [
 "VADER is smart, handsome, and funny.",       # positive sentence
 "VADER is smart, handsome, and funny!",    # punctuation emph., adj. intensity
 "VADER is very smart, handsome, and funny.", # booster words
 "VADER is VERY SMART, handsome, and FUNNY.", # emphasis for ALLCAPS
 "VADER is VERY SMART, handsome, and FUNNY!!!",# many signals, adj. intensity
 "VADER is VERY SMART, really handsome, and INCREDIBLY FUNNY!!!", # booster
     words & punctuation make this close to ceiling for score
 "The book was good.",                         # positive sentence
 "The book was kind of good.",                 # qualified pos., adj. intensity
 "The plot was good, but uncompelling characters and the dialog is not great.",
                                               # mixed negation sentence
 ...
]
paragraph = "It was one of the worst movies I've seen, despite good reviews. \
Unbelievably bad acting!! Poor direction. VERY poor production. \
The movie was bad. Very bad movie. VERY bad movie. VERY BAD movie. VERY BAD
    movie!"
```

Challenges of Natural Language Processing

- Biggest challenge in NLP is simply that human language is ambiguous
- Even humans struggle to analyze and classify human language correctly.
  - Sarcasm, humor, . . .
- Natural language processing and powerful machine learning algorithms (often multiple used in collaboration) are improving, and bringing order to the chaos of human language

- SMMRY: Summarize my text in [7] sentences.
- TextBlob is a Python library with a simple interface to perform a variety of NLP tasks. Built on the shoulders of NLTK and another library called Pattern, it is intuitive and user-friendly, which makes it ideal for beginners. Learn more about how to use TextBlob and its features
- Natural Language Toolkit (NLTK) is a suite of libraries for building Python programs (Bird et al. (2009)). It is the most popular Python library for NLP, has a very active community behind it, and is often used for educational purposes
- SpaCy is a free open-source library for advanced natural language processing in Python. It has been specifically designed to build NLP applications that can help you understand large volumes of text
- MonkeyLearn is a SaaS platform that lets you build customized NLP models to perform tasks like sentiment analysis and keyword extraction

NLTK vs. SPACY

- Choosing one package over another is often not clear-cut as it depends on the circumstances. Your choice could depend on:
  - Focus. NLTK sees things holistically, while spaCy is known for its granular approach. NLTK == used to develop complex NLP functions via different stemming libraries. Used by researchers to build something from scratch. spaCy == single stemmer and good fit for app builders
  - Processing. NLTK takes strings as input and returns lists of them as output. spaCy is object-oriented: every function returns objects as output. With NLTK, developers have to check out the documentation on a regular basis, while spaCy allows for easy exploration.
  - Performance. NLTK considerably slower than spaCy: the latter was written in Cython from scratch. Also, spaCy exceeds NLTK with regard to part-of-speech tagging and word tokenization.
- Nice spaCy and NLTK tutorials

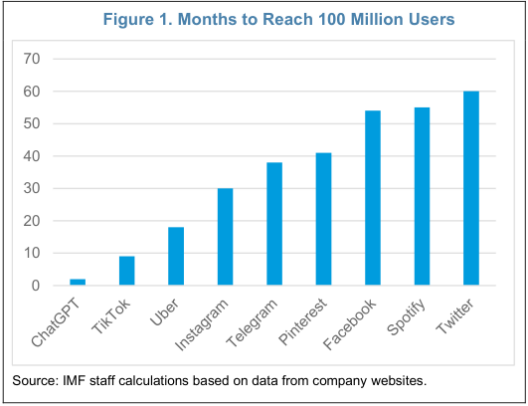
What is BERT?

Bi-directional **E**ncoder **R**epresentations from **T**ranformers

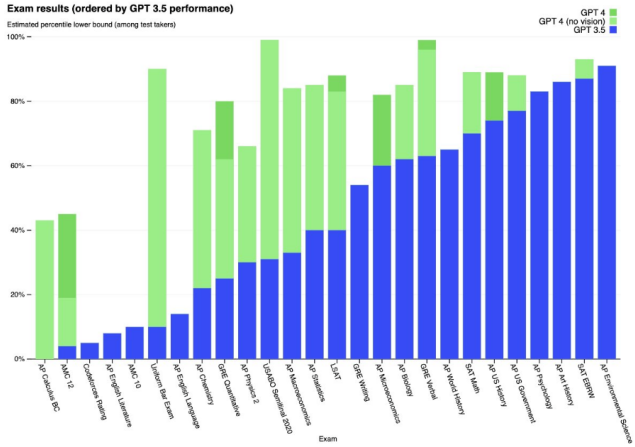
- BERT was trained on 2500M words in Wikipedia and 800M from books.
- BERT reads the sentence from left to right but also from right to left
- BERT encodes the sentence to a vector so that we can work with It
- BERT uses an attention mechanism to use the context of the use of a word
- Other language models Generative Pre-trained Transformers: GPT-3 and OPT-175B

And then there was ChatGPT

- Incredibly fast pick-up
- ChatGPT has an uncanny performance
- Especially useful for writing code and generating boilerplate text
- It passed the Bar and passed the US medical licensing exam
- New version of underlying engine (GPT-4) released 14/3/2023



GPT performance at exams



Source: Eloundou and Manning 2023

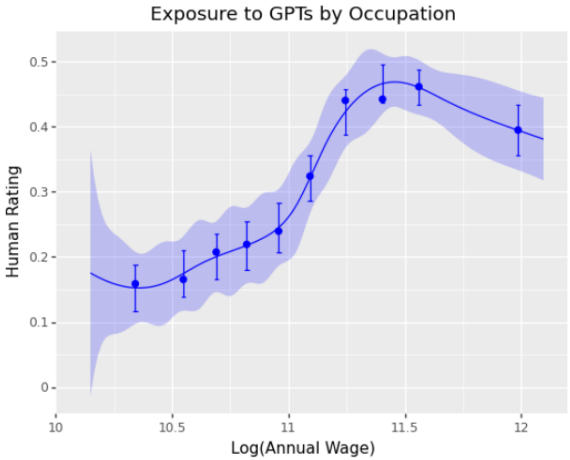
DataScience Hub

Effects and concerns

1. Effects on employment: will change types of jobs not yet touched by automation
2. Hallucinating: ChatGPT has the tendency to authoritatively make up stuff
3. Bias: it's trained on what was on the internet until 2021. The internet is not representative or inclusive
4. Fake news: makes it very cheap and convincing to generate disinformation. See Frankfurt's article "On Bullshit" (Frankfurt (1986))
5. Boring: it generates the most probable text. This is often middle of the road

See Matt Turck's blog for further discussion

Not all jobs are affected equally



Source: Eloundou and Manning 2023

Overview of LLMs

Model	Organization	Size	Release	Achievements and NLP Tasks
BERT	Google AI	340 ml	10-2018	Various NLP tasks. GLUE 86.5 – Question answering, NLI
Turing-NLG	Microsoft	17 bn	2-2020	Used transformer – Q.A., NLI, text summarization GLUE score: 92.8
Megatron-TNLG	Google AI	530 bn	10-2021	SuperGLUE – NLP understanding, generation GLUE score: 92.6
LaMDA	Google AI	137 bn	5-2022	C4 benchmark – Conversational AI, Q.A., creative text GLUE score: 93.4
Blender	Blender Inst.	137 bn	5-2022	C4 benchmark – Q.A., NLI, creative writing GLUE score: 92.9
Jurassic-1 Jumbo	Google AI	1.75 tr	6-2022	GLUE benchmark – NLP understanding, generation GLUE score: 94
WuDao 2.0	Beijing Ac. of AI	1.75 tr	6-2022	GLUE benchmark – NLP understanding, machine translation GLUE score: 94.2

Overview of LLMs (cont.)

Model	Organization	Size	Release	Achievements and NLP Tasks
GPT-3	OpenAI	175 bn	11-2022	Variety of NLP tasks – NLP, including machine translation, text summarization, and question answering GLUE score: 80.3
GPT-4	OpenAI	175 bn	3-2023	Better than GPT-3 on complex tasks, though slower and more expensive – Similar to GPT-3 GLUE score: 93.3
Claude	Anthropic	137 bn	3-2023	GLUE benchmark for NL understanding – Good on both broad and complex tasks, more balanced than GPT-4 GLUE score: 92.2.
Pi Inflection	Anthropic	137 bn	6-2023	GLUE benchmark for NL understanding – Excellent on both broad and complex tasks, comparable with GPT-4 GLUE score: 92.8

Getting a sentiment is really easy


```
[1]: import openai
     import os

[2]: openai.api_key = os.getenv("OPENAI_API_KEY")

     def get_response_to_prompt(prompt):
         response = openai.ChatCompletion.create(model="gpt-3.5-turbo", messages=[{"role":"user","content":prompt}], tempera
         return response.choices[0].message["content"]

[3]: prompt = """Classify the text below, delimited by three dashes (-), as having either a positive or negative sentiment.

     ---
     I had a fantastic time at Stanford GSB! Learned a lot and also made great new friends!
     ---
     """

[4]: response = get_response_to_prompt(prompt)
     print(response)

     positive sentiment

[ ]:
```

DataScience Hub

Basic ChatGPT

- Fundamentally ChatGPT is trying to produce a “reasonable continuation” of whatever text it’s got
- Complete the following: The best thing about AI is its ability to ...

- Fundamentally ChatGPT is trying to produce a “reasonable continuation” of whatever text it’s got
- Complete the following: The best thing about AI is its ability to ...

learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

- Fundamentally ChatGPT is trying to produce a “reasonable continuation” of whatever text it’s got
- Complete the following: The best thing about AI is its ability to ...

learn	4.5%
predict	3.5%
make	3.2%
understand	3.1%
do	2.9%

- Interestingly, picking highest probabilities leads to “flat” text. So using an arbitrary temperature parameter we pick a lower ranked word.

Basic ChatGPT (cont.)

- With temperature = 0, text produced becomes repetitive and boring

The best thing about AI is its ability to learn from experience. It's not just a matter of learning from experience, it's learning from the world around you. The AI is a very good example of this. It's a very good example of how to use AI to improve your life. It's a very good example of how to use AI to improve your life. The AI is a very good example of how to use AI to improve your life. It's a very good example of how to use AI to

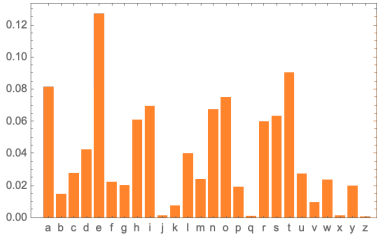
Basic ChatGPT (cont.)

- With temperature = 0, text produced becomes repetitive and boring
- Temperature at 0.8 (and some tweaks in GPT3) it is more Interesting

The best thing about AI is its ability to learn and develop over time, allowing it to continually improve its performance and be more efficient at tasks. AI can also be used to automate mundane tasks, allowing humans to focus on more important tasks. AI can also be used to make decisions and provide insights that would otherwise be impossible for humans to figure out.

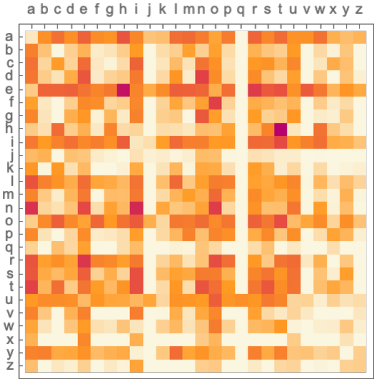
Where do the probabilities come from?

- We could start with the probabilities of letters in the English language or ...



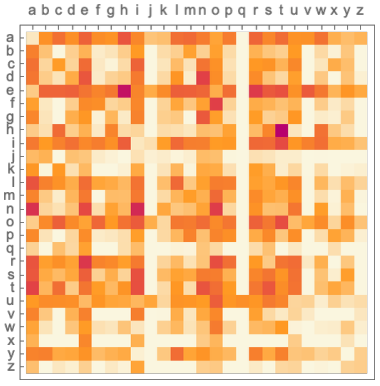
Where do the probabilities come from?

- We could start with the probabilities of letters in the English language or ...
- ... the combinations of letters (n-gram)



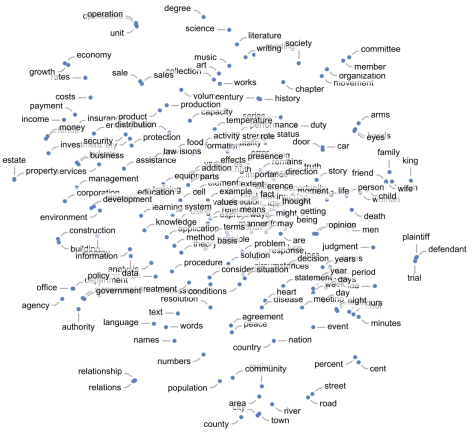
Where do the probabilities come from?

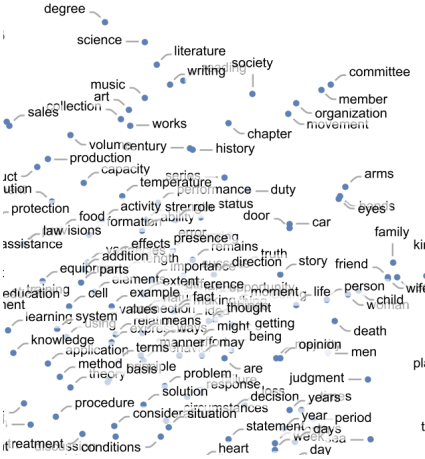
- We could start with the probabilities of letters in the English language or ...
- ... the combinations of letters (n-gram)
- Or take it up one level and look at words ...
- ... with 40,000 common words the number of possible 2-grams of words is already 1.6 bn and ...
- ... the number of possible 3-grams is 60 tr so estimating the probabilities of sentences is impossible
- ChatGPT makes it work with 175 bn weights



DataScience Hub

Embeddings

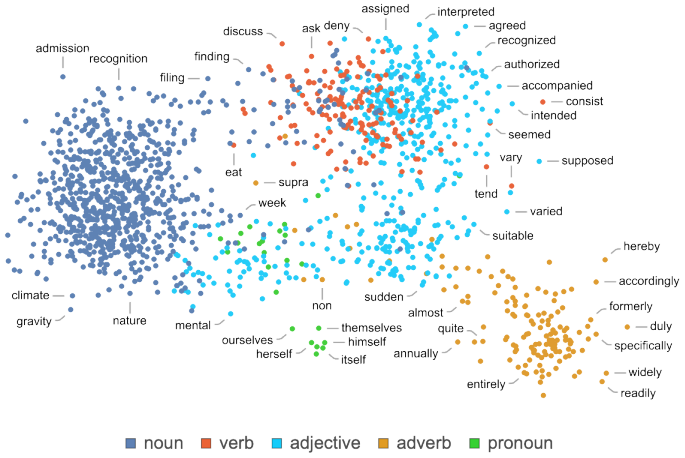




Source: Wolfram

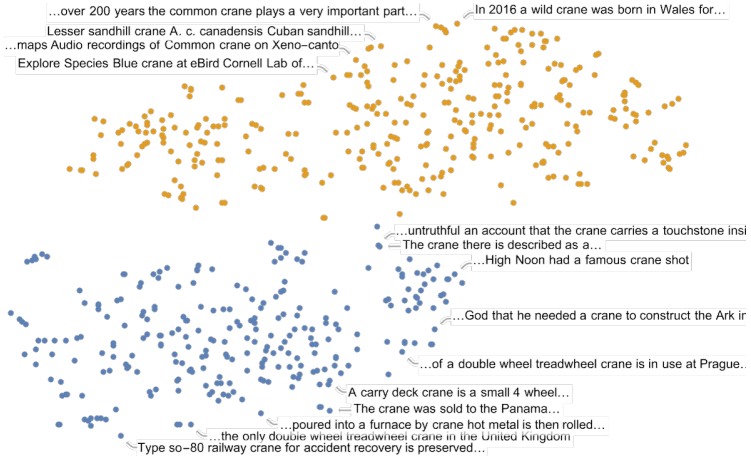
DataScience Hub

Embeddings



Source: Wolfram

Embeddings



Source: Wolfram

DataScience Hub

Summary

In this lecture we covered:

1. some examples of Natural Language Processing (NLP)
2. a discussion of the different building blocks needed (e.g. tokenization, lemmatization, etc.)
3. how to use these methods to extract the right information from textual input

Further reading

- Solid textbook: (Speech and Language Processing n.d.)
- Natural Language Processing (NLP): What Is It and How Does it Work?
- Ivan Habernal
- BERT NLP Model Explained for Complete Beginners
- Seminal paper on why transformer models are better than LSTMs (Vaswani et al. 2017)

References I

AlAjmi, K., Deodore, J., Khan, A., & Moriya, K. (2023). Predicting the Law: Artificial Intelligence Findings from the IMF's Central Bank Legislation Database. IMF Working Paper, 2023/241. Retrieved November 19, 2023, from https://www.imf.org/en/Publications/WP/Issues/2023/11/18/Predicting-the-Law-Artificial-Intelligence-Findings-from-the-IMFs-Central-Bank-Legislation-541619

Baker, S. R., Bloom, N., & Davis, S. J. (2016). Measuring Economic Policy Uncertainty. The Quarterly Journal of Economics, 131(4), 1593–1636.

Baumgärtner, M., & Zahner, J. (2022). Whatever it Takes to Understand a Central Banker-Embedding their Words Using Neural Networks. 1–52. www.uni-marburg.de/fb02/MACIE

Bholat, D., Hansen, S., Santos, P., & Schonhardt-Bailey, C. (2015). Text Mining for Central Banks. Centre for Central Banking Studies, 33.

Bird, S., Loper, E., & Klein, E. (2009). Natural Language Processing with Python. O'Reilly Media Inc.. O'Reilly.

References II

- Chappell, H. W., Havrilesky, T. M., & McGregor, R. R. (1997). Monetary policy preferences of individual FOMC members: A content analysis of the memoranda of discussion. Review of Economics and Statistics, 79(3), 454–460.
- Das, S. R. (2014). Text and Context: Language Analytics in Finance (Vol. 8) [Publication Title: Foundations and Trends® in Finance Issue: 3 ISSN: 1567-2395].
- Dim, C., Koerner, K., Wolski, M., & Zwart, S. (2021). News-Implied Sovereign Default Risk. SSRN Electronic Journal.
- Eloundou, T., & Manning, S. (2023). GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models. Retrieved October 8, 2023, from https://ar5iv.labs.arxiv.org/html/2303.10130
- Engle, R., Giglio, S., Lee, H., Kelly, B., Stroebel, J., & Stern, N. (2020). Hedging Climate Change News. Review of Financial Studies, 33(3), 1184–1216.
- Frankfurt, H. (1986). On Bullshit. http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:On+Bullshit#0

References III

García, D., Hu, X., & Rohrer, M. (2023). The colour of finance words. Journal of Financial Economics, 147(3), 525–549.

Hassan, T. A., Hollander, S., Lent, L. V., & Tahoun, A. (2020). Firm-level political risk: Measurement and effects. The Quarterly Journal of Economics, 2135–2202.

Kazinnik, S., Scid, D., & Wu, J. (2021). News and Networks : Using Text Analytics to Assess Bank Networks During COVID-19 Crisis.

Li, K., Mai, F., Shen, R., & Yan, X. (2021). Measuring Corporate Culture Using Machine Learning. The Review of Financial Studies, 34, 3265–3315.

Liu, Y.-W., Liu, L.-C., Wang, C.-J., & Tsai, M.-F. (2018). RiskFinder: A Sentence-level Risk Detector for Financial Reports. Proceedings of NAACL-HLT 2018: Demonstrations, 81–85. Retrieved May 26, 2022, from https://cfda.csie.

Loughran, T., & Mcdonald, B. (2019). Textual Analysis in Finance. 1–23.

References IV

- Speech and Language Processing. (n.d.). Retrieved March 15, 2023, from https://web.stanford.edu/~jurafsky/slp3/
- Tetlock, P. C. (2007). Giving Content to Investor Sentiment: The Role of Media in the Stock Market [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.2007.01232.x]. The Journal of Finance, 62(3), 1139–1168.
- Van Dijk, D., & De Winter, J. (2023). Nowcasting GDP using tone-adjusted time varying news topics: Evidence from the financial press. DNB Working Paper, 766.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need [arXiv:1706.03762 [cs]].