From Statistics to Machine Learning

Text Alalysis

Parantapa Goswami

Viseo R&D
Grenoble, France
parantapa.goswami@viseo.com

July 9, 2017

Outline

Unstructured data

- Data that does not have a pre-defined model or structure
- Typically contains free text in form of natural language data
- Estimated $80 - 90\%$ of ‘usable’ information in unstructured form

Example: abstract of a research article

Wegener’s granulomatosis presenting during first trimester of pregnancy. We describe a case of Wegener’s granulomatosis in a lady who presented acutely with pulmonary hemorrhage, fever and breathlessness during her early pregnancy. She responded well to aggressive medical treatment.

Unstructured data

- Data that does not have a pre-defined model or structure
- Typically contains free text in form of natural language data
- Estimated 80 – 90% of 'usable' information in unstructured form

Example: excerpt of a medical record

Physical Exam
General Appearance: well developed, well nourished, no acute distress
Eyes: conjunctiva and lids normal, PERRLA, EOMI, fundi WNL
Ears, Nose, Mouth, Throat: TM clear, nares clear, oral exam WNL
Respiratory: clear to auscultation and percussion, respiratory effort normal
Cardiovascular: regular rate and rhythm, S1-S2, no murmur, rub or gallop, no bruits, peripheral pulses normal and symmetric, no cyanosis, clubbing, edema or varicosities
Skin: clear, good turgor, color WNL, no rashes, lesions, or ulcerations
Problems (including changes): Blood pressure is lower. Feet are inspected and there are no callouses, no compromised skin. No vision complaints.

Unstructured data

- Data that does not have a pre-defined model or structure
- Typically contains free text in form of natural language data
- Estimated $80-90\%$ of 'usable' information in unstructured form

Challenges

- Data access: storing and accessing unstructured data is not trivial in comparison to their structured numeric counterparts
- Human issue: text differs depending on the author
- Language issue: text can be written in any of the 6909 existing languages in the world

Importance of Unstructured Data

- Different data types (articles, patents, reports, clinical records etc..)
- Easy for direct human interpretation
- Databases only cover a fraction of biological context from the literature
- Contextual information of experimental results (cell line, tissue, conditions)
- User demands of better information access beyond keyword searches
- Rapid growth of information, manual information extraction not efficient

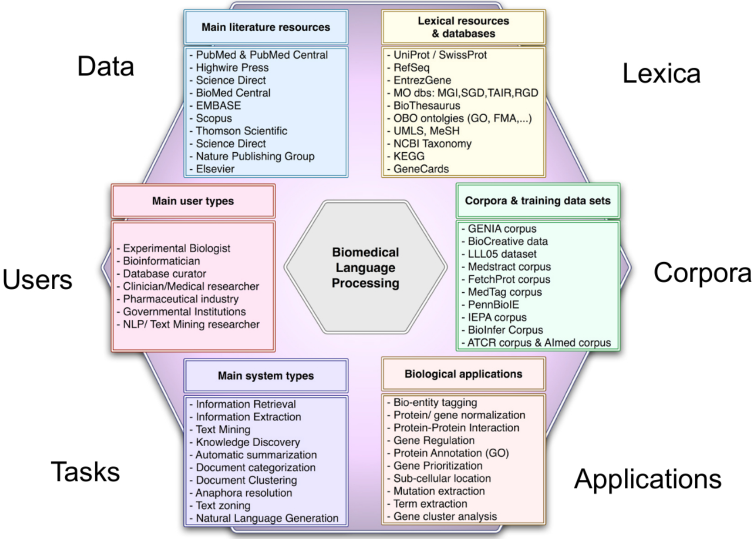
Goals

Text Mining

Text mining is the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. – Wikipedia

- Discovery and extraction of knowledge from unstructured data
- Extraction of non-trivial information or new information from natural language data collections
- Applications integrating various natural language processing components (such as document retrieval, classification and recognition of entities).

Text Mining for Biomedical

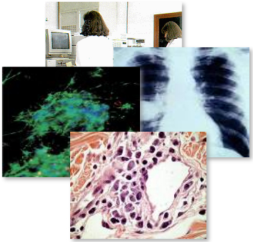


Slide from Text Mining course by M. Krallinger and F. Leitner

User Groups

Biology

- Define the biological question
- Select the actual target being studied
- Extract information relevant for experimental set up
- Locate relevant resources
- Essential to understand and interpret the resulting data
- Draw conclusions about new discoveries
- Communicated to the scientific community using publications in peer-reviewed journals



Clinics

- Resource for clinical decision support in evidence-based clinical practice
- Useful information for diagnostic aids

Pharma

- Drug discovery and target selection
- Identifying adverse drug effect
- Competitive intelligence and knowledge management

Funding

- Global view of the current research state & monitor trends to ensure optimal resource allocation

Publ.

- Find domain experts for specific topics for the peer-review process & detecting potential cases of plagiarism

Slide from Text Mining course by M. Krallinger and F. Leitner

1. Motivation
2. Linguistic Text Preprocessing
3. Text Representation
4. Applications

Preprocessing

- Linguistic preprocessing is a series of techniques that transforms raw text into an computer understandable format
- Goal of preprocessing is to construct a *vocabulary*: a computer understandable and efficiently accessible format to store the input raw text
- Preprocessing typically includes:
  - Tokenization
  - Normalization
  - Filtering
  - Indexing

Tokenization

- Given a character sequence and a defined document unit, tokenization is the task of chopping up the character sequence into pieces, called *tokens*
- It also removes certain characters: such as punctuation.

Example

Input: Six patients were hospitalized for priapism, and four patients were treated with a shunting procedure.

Output: $\boxed{\text{Six}}$ $\boxed{\text{patients}}$ $\boxed{\text{were}}$ $\boxed{\text{hospitalized}}$ $\boxed{\text{for}}$ $\boxed{\text{priapism}}$ $\boxed{\text{and}}$ $\boxed{\text{four}}$ $\boxed{\text{patients}}$ $\boxed{\text{were}}$ $\boxed{\text{treated}}$ $\boxed{\text{with}}$ $\boxed{\text{a}}$ $\boxed{\text{shunting}}$ $\boxed{\text{procedure}}$

- After tokenization we have 15 **tokens** but only 13 **types**
- **Token** is an instance of a sequence of characters
- **Type** is the class of all tokens containing the same character sequence

Tokenization: Challenges

- Apostrophe
  - `Parkinson's disease` →
    $\boxed{\text{Parkinson}}$ $\boxed{\text{s}}$ OR $\boxed{\text{Perkinsons}}$ OR $\boxed{\text{Parkinson's}}$
- Hyphenated sequence
  - `state-of-the-art`: break it up or not?
- Multiword tokens
  - `San Francisco`: one token or two tokens?

Tokenization: Language Issues

A good tokenization method must take into account the language of the text.

- French has variant use of apostrophe (*l'ensemble*) or hyphens (*donne-moi*)
- German noun compounds are not segmented
  e.g. *Lebensversicherungsgesellschaftsangestellter*: life insurance company employee
- Chinese and Japanese have no spaces between words
- Arabic (or Hebrew) is written right to left, but with certain items like numbers written left to right

Tokenization: Tools

- For European languages, different software tools are commercially available
- Some of these tools performs tokenization in an intelligent manner: for example associating the words of a sentence with grammatical rules.
- TreeTagger is one such open source software
  `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`

Normalization

- Token normalization is the process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens

Example

`window` $\leftarrow$ `Windows, windows, Window, window`

`USA` $\leftarrow$ `U.S.A, USA, usa`

- The most standard way to normalize is to implicitly create equivalence classes, which are normally named after one member of the set
- Normalization methods are broadly categorized into two types: *surface* or *textual* normalization and *linguistic* normalization

Normalization: Textual

Textual normalization transforms the words of the same family in their canonical form by making some superficial changes on the sequences of characters of these words.

- **Punctuations:** One of the basic rule is to remove all dots and hyphens from the words. It is particularly useful for acronyms, e.g. U.S.A→USA.
- **Case of letters:** A classical strategy is to make the letters lower case. However, it may not be suitable for acronyms and certain proper names.
- **Accents:** The rule applied generally consists of removing the diacritics on all the words. e.g. résumé → resume.

Normalization: Linguistic

Linguistic normalization uses knowledge on underlying languages and linguistic rules to transform a word to its canonical form.

Lemmatization

- Reduce variant forms of words to a base form
- Example:
  - `am, are, is` → `be`
  - `car, cars, car's, cars'` → `car`
- Lemmatization implies doing "proper" reduction to dictionary headword form
- Most commonly used text processing tools (e.g. Python NLTK) has lemmatizor for different languages

Normalization: Linguistic

Linguistic normalization uses knowledge on underlying languages and linguistic rules to transform a word to its canonical form.

Stemming

- Reduce words to their "roots" or "stems"
- Performs *crude affix chopping* based on some predefined rules
- Example of typical rules
  - `sses` → `ss`, e.g. processes → process
  - `ies` → `i`, e.g. berries → berri
  - `tional` → `tion`, e.g. national → nation
- Most used stemming algorithm for English: Porter Stemmer
- Snowball Stemmer (`http://snowballstem.org/`): provides stemmer for 18 different languages.

Filtering

Filtering removes words that tend to be present with high frequency in all documents in a collection and that provide little information about the content of a document.

Example

- In a collection of medical records, the word "patient" will occur multiple times in all the documents.
- In a collection of automobile documents, the word "car" will occur multiple times in all the documents.

Common strategy:

1. to set a threshold on number of times a word can occur in a collection
2. remove all words with higher occurrence than the threshold

Filtering: Stop-Words

Stop words usually refer to the most common words in a language

- They have little semantic content
- There are a lot of them: about 30% to 35% of most frequent words in a collection are only top 30 words

English stop-words

Articles	a, an, the
Pronouns	he, she, him, her etc.
Prepositions	in, by, with etc.
Conjunctions	and, but, while etc.
Other common words	yes, no, very etc.

Filtering: Stop-Words

Stop words usually refer to the most common words in a language

- They have little semantic content
- There are a lot of them: about 30% to 35% of most frequent words in a collection are only top 30 words

Stop-list

- A list of stop-words used by text processing tools
- There exist no *universal stop-list*: however, the existing lists are mostly accepted by the community and used in practice

Zipf's Law

Zipf's Law specifies that the frequency of occurrence of a word w in a text collection is inversely proportional to the rank of the word in the frequency list.

$$Freq_w \propto \frac{1}{Rank_w}$$

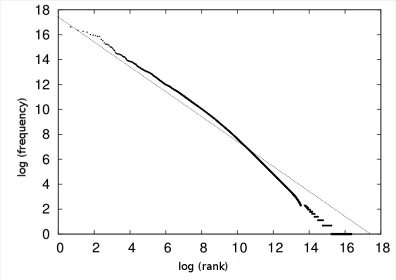
- Empirical law formulated using mathematical statistics
- Easily observed by plotting the data on a log-log graph:
  - x-axes: log(rank order)
  - y-axes: log(frequency)
  - should yield a straight line with a negative slope

Zipf's Law

Zipf's Law specifies that the frequency of occurrence of a word w in a text collection is inversely proportional to the rank of the word in the frequency list.

$$Freq_w \propto \frac{1}{Rank_w}$$

Case study: French Wikipedia



rank	word	frequency
1	de	36,875,868
2	la	16,565,726
3	le	12,639,034
4	et	11,587,487
5	en	10,885,221
6	l'	8,937,203
⋮	⋮	⋮

Bag of Words

- A simplifying representation used in natural text processing
- A text (such as a sentence or a document) is represented as the bag of its words:
  - disregards: grammar, word ordering
  - maintains: multiplicity

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

Vector Space Model

- The Vector Space Model defines a representation of documents commonly used in various tasks of accessing information from text.

Vocabulary

Vocabulary is the set of unique words/terms present in the collection.

- Set of **types**

- In Vector Space Model:
  - each document d of a collection $\mathscr{C}$ is associated with a vector $\vec{d}$
  - the dimension of vector $\vec{d}$ to the size of the vocabulary
- The vector space constructed is a space of terms in which each dimension is associated with a term of the collection.

Vector Space Model: Count Matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
`Anthony`	157	73	0	0	0	1
`Brutus`	4	157	0	2	0	0
`Caesar`	232	227	0	2	1	0
`Calpurnia`	0	10	0	0	0	0
`Cleopatra`	57	0	0	0	0	0
`mercy`	2	0	3	8	5	8
`worser`	2	0	1	1	1	5
...						

Each document is now represented as a count vector $\in \mathbb{N}^{|V|}$.
$V \implies$ the vocabulary of the collection under consideration

Vector Space Model: Count Matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
`Anthony`	157	73	0	0	0	1
`Brutus`	4	157	0	2	0	0
`Caesar`	232	227	0	2	1	0
`Calpurnia`	0	10	0	0	0	0
`Cleopatra`	57	0	0	0	0	0
`mercy`	2	0	3	8	5	8
`worser`	2	0	1	1	1	5
...						

Each document is now represented as a count vector $\in \mathbb{N}^{|V|}$.

$V \implies$ the vocabulary of the collection under consideration

Term Frequency

- The term frequency $\mathrm{tf}_{t,d}$ of term t in document d is defined as the number of times that t occurs in d.
- A term in a document with tf = 10 occurrences is more important than a term with tf = 1 occurrence in that document.
- The frequency of the term in the collection...
  - Some terms are rare (e.g. `EPISTEMOPHOBIA`).
  - Some terms are frequent (e.g. `PATIENT`)
  - Rare terms are more informative than frequent terms.
  - Rare terms should have higher weights than frequent terms.

Document Frequency

- We want:
  - higher weights for rare terms like `EPISTEMOPHOBIA`.
  - lower (but positive) weights for frequent terms like `PATIENT`.
- The document frequency df_t of a term t is the number of documents in the collection that t occurs in.
- Rare terms will have lower df than frequent terms.
- df_t is an inverse measure of the informativeness of term t.

Inverse Document Frequency

- Inverse document frequency idf_t of term t is:

$$Idf_t = \log \frac{N}{df_t}$$

(N is the number of documents in the collection.)

- idf_t is a measure of the informativeness of the term.
- In the phrase "epistemophobia patient", idf weighting increases the relative weight of `EPISTEMOPHOBIA` and decreases the relative weight of `PATIENT`.

TF-IDF Scoring

- The tf-idf weight of a document d w.r.t a query term t is the product of its $\text{tf}_{t,d}$ and idf_d.

$$score_{tf-idf}(t,d) = (tf_{t,d}).\left(\log \frac{N}{df_t}\right)$$

- The tf-idf score:
  - increases with the number of occurrences of t within d. (term frequency)
  - increases with the rarity of the term in the collection. (inverse document frequency)
- Tf-idf weight of a term in a document represents the importance or informativeness of the term in the document

Vector Space Model: Weight Matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Anthony	157	73	0	0	0	1
Brutus	4	157	0	2	0	0
Caesar	232	227	0	2	1	0
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	8	5	8
worser	2	0	1	1	1	5
...						

Vector Space Model: Weight Matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
`Anthony`	5.25	3.18	0.0	0.0	0.0	0.35
`Brutus`	1.21	6.10	0.0	1.0	0.0	0.0
`Caesar`	8.59	2.54	0.0	1.51	0.25	0.0
`Calpurnia`	0.0	1.54	0.0	0.0	0.0	0.0
`Cleopatra`	2.85	0.0	0.0	0.0	0.0	0.0
`mercy`	1.51	0.0	1.90	0.12	5.25	0.88
`worser`	1.37	0.0	0.11	4.15	0.25	1.95
...						

Each document is now represented as a real-valued vector of tf-idf weights $\in \mathbb{R}^{|V|}$.

$V \implies$ the vocabulary of the collection under consideration

Vector Space Model: Weight Matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Anthony	5.25	3.18	0.0	0.0	0.0	0.35
Brutus	1.21	6.10	0.0	1.0	0.0	0.0
Caesar	8.59	2.54	0.0	1.51	0.25	0.0
Calpurnia	0.0	1.54	0.0	0.0	0.0	0.0
Cleopatra	2.85	0.0	0.0	0.0	0.0	0.0
mercy	1.51	0.0	1.90	0.12	5.25	0.88
worser	1.37	0.0	0.11	4.15	0.25	1.95
...						

Each document is now represented as a real-valued vector of tf-idf weights $\in \mathbb{R}^{|V|}$.

$V \implies$ the vocabulary of the collection under consideration

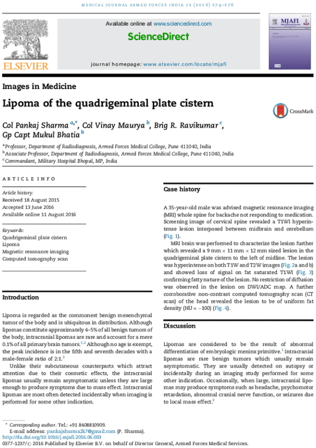
Alternative Representations

- Each document is now represented as a real-valued vector of tf-idf weights $\in \mathbb{R}^{|V|}$.
- Vectors constructed in this way are **sparse**
  - Very inefficient while accessing
  - Lots of computer memory is wasted to store a large number of zeros
- Alternatives:
  - Various mathematical computation tools (e.g. Python Numpy, Matlab) has memory efficient methods to store sparse matrix
  - Inverted index...

1. Motivation
2. Linguistic Text Preprocessing
3. Text Representation
4. Applications

Text Classification

- Assigning categories to a piece of text



MEDICAL JOURNAL ARMED FORCES INDIA 72 (2016) S74–S76

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/mjafi

Images in Medicine

Lipoma of the quadrigeminal plate cistern

Col Pankaj Sharma [a,*], Col Vinay Maurya [b], Brig R. Ravikumar [c], Gp Capt Mukul Bhatia [b]

[a] Professor, Department of Radiodiagnosis, Armed Forces Medical College, Pune 411040, India
[b] Associate Professor, Department of Radiodiagnosis, Armed Forces Medical College, Pune 411040, India
[c] Commandant, Military Hospital Bhopal, MP, India

ARTICLE INFO

Article history:
Received 18 August 2015
Accepted 13 June 2016
Available online 11 August 2016

Keywords:
Quadrigeminal plate cistern
Lipoma
Magnetic resonance imaging
Computed tomography scan

Introduction

Lipoma is regarded as the commonest benign mesenchymal tumor of the body and is ubiquitous in distribution. Although lipomas constitute approximately 4–5% of all benign tumors of the body, intracranial lipomas are rare and account for a mere 0.1% of all primary brain tumors.[1,2] Although no age is exempt, the peak incidence is in the fifth and seventh decades with a male-female ratio of 2:1.[3]

Unlike their subcutaneous counterparts which attract attention due to their cosmetic effects, the intracranial lipomas usually remain asymptomatic unless they are large enough to produce symptoms due to mass effect. Intracranial lipomas are most often detected incidentally when imaging is performed for some other indication.

Case history

A 35-year-old male was advised magnetic resonance imaging (MRI) whole spine for backache not responding to medication. Screening image of cervical spine revealed a T1WI hyperintense lesion interposed between midbrain and cerebellum (Fig. 1).

MRI brain was performed to characterize the lesion further which revealed a 9 mm × 11 mm × 12 mm sized lesion in the quadrigeminal plate cistern to the left of midline. The lesion was hyperintense on both T1W and T2W images (Fig. 2a and b) and showed loss of signal on fat saturated T1WI (Fig. 3) confirming fatty nature of the lesion. No restriction of diffusion was observed in the lesion on DWI/ADC map. A further corroborative non-contrast computed tomography scan (CT scan) of the head revealed the lesion to be of uniform fat density (HU = −100) (Fig. 4).

Discussion

Lipomas are considered to be the result of abnormal differentiation of embryologic meninx primitive.[1] Intracranial lipomas are rare benign tumors which usually remain asymptomatic. They are usually detected on autopsy or incidentally during an imaging study performed for some other indication. Occasionally, when large, intracranial lipomas may produce symptoms such as headache, psychomotor retardation, abnormal cranial nerve function, or seizures due to local mass effect.[2]

* Corresponding author. Tel.: +91 8408810909.
E-mail address: pankajsharma2k7@gmail.com (P. Sharma).
http://dx.doi.org/10.1016/j.mjafi.2016.06.003

MeSH Subject Categories:

- Antogonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- ...

Text Classification: Definition

- Input:
  - a document d
  - a fixed set of categories or classes $C = \{c_1, c_2, \ldots\}$
- Output: a predicted class $c \in C$ for document d

Text Classification: Rule Based Methods

- Manually crafted rules based on combination of words
  - *drug therapy* if these two words co-occur significant number of times
- Accuracy can be high
  - if rules are carefully crafted and refined by experts
- Building and maintaining such rules are expensive and infeasible for large collections

Text Classification: Machine Learning Methods

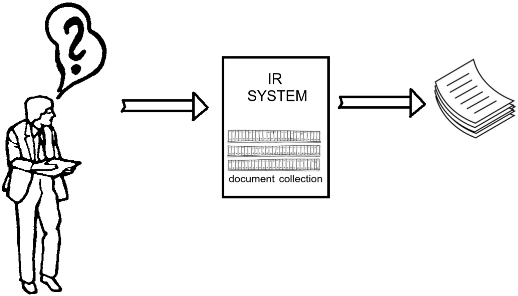
- Input:
  - a document d
  - a fixed set of categories or classes $C = \{c_1, c_2, \ldots\}$
  - a training set of some hand-labeled documents $(d_1, c_1), (d_2, c_2), \ldots$
- Output: a learned classifier $\lambda : d \to c$

Text Classification: Machine Learning Methods

- There exist lots and lots of classifiers:
  - Naive Bayes
  - Logistic regression
  - Support-vector machines
  - Neural networks
  - ...
- Machine learning tools:
  - Python ScikitLearn
  - WEKA
  - MATLAB
  - ...

Information Retrieval

Information retrieval (IR) is finding documents of an unstructured nature (usually text) that satisfies an query from within large collections.



- A query is typically few words
  - can be considered as a mini-document

Information Retrieval: using Vector Space Model

- Each document is represented as a real-valued vector of tf-idf weights $\in \mathbb{R}^{|V|}$
- To score documents w.r.t queries:
  1. similar representation can be obtained for a given query
  2. rank documents according to their similarity to the query.
- Cosine similarity is used to measure similarity between query vector $\vec{q}$ and document vector $\vec{d}$.

$$score_{vspace}(q, d) = \cos(\vec{q}, \vec{d}) = \frac{\vec{q}.\vec{d}}{|\vec{q}|.|\vec{d}|}$$

An unbounded Potential

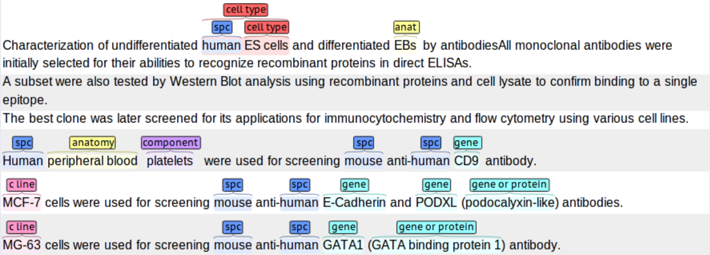
Information Extraction (IE)

- Find and understand limited relevant parts of text
- Gather information from many pieces of text
- Produce a structured representation of relevant information
  - relations: drug-gene, sub-cellular localization etc.

An unbounded Potential

Named Entity Recognition (NER)

Find and **classify** names in text



It's only the beginning...

Resources

- Git-hub: `https://github.com/parantapag/IBD4Health2017`
- Above link is also provided at INDICO portal.