



Machine Learning

Natural Language Processing

Karol Przystalski

April 20, 2022

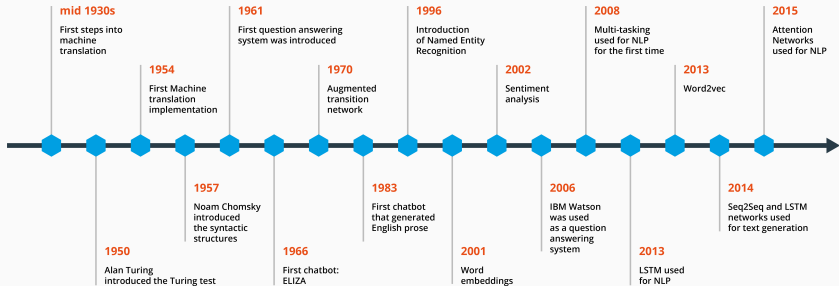
Department of Information Technologies, Jagiellonian University

Agenda

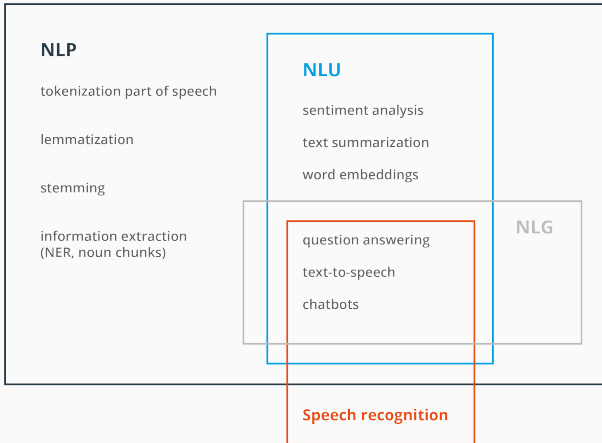
1. Introduction
2. Use cases
3. Text processing
4. Natural Language Processing
5. Text understanding
6. Chatbots
7. Text generation
8. Quality metrics

Introduction

Timeline



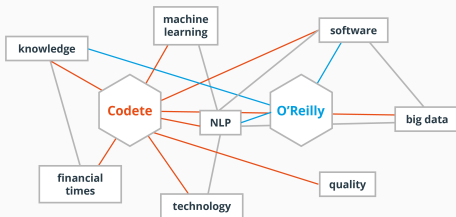
Taxonomy



Use cases

There are many use cases of NLP methods, just to mention a few:

- term relationship,



There are many use cases of NLP methods, just to mention a few:

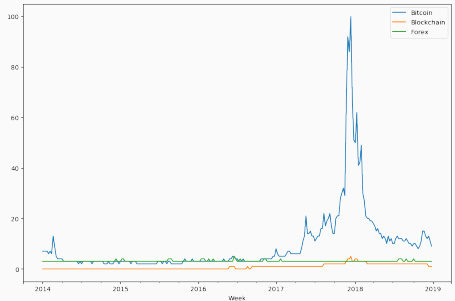
- term relationship,
- text summarization,



Brexit: UK and UE negotiations failed

There are many use cases of NLP methods, just to mention a few:

- term relationship,
- text summarization,
- time series analysis,



There are many use cases of NLP methods, just to mention a few:

- term relationship,
- text summarization,
- time series analysis,
- intent recognition,



There are many use cases of NLP methods, just to mention a few:

- term relationship,
- text summarization,
- time series analysis,
- intent recognition,
- sentiment analysis,



Financial Times

Are we facing a confidence gap?
While women outperform men
in school and are entering
the workforce in record numbers,
half of female managers in the UK
reported self doubt about their job
performance and careers



Are we facing a confidence gap?
While women outperform men
in school and are entering
the workforce in record numbers,
half of female managers in the UK
reported self doubt about their job
performance and careers

There are many use cases of NLP methods, just to mention a few:

- term relationship,
- text summarization,
- time series analysis,
- intent recognition,
- sentiment analysis,
- text generation.

What do we want?

Chatbots!

When is the next
training on chatbots?

Sorry, I don't
understand your
request

Text processing

Regular expressions in Python for string comparison

Regular expressions in Python are almost the same as in any other programming languages. We can use regex methods to:

- `search` – finds only the first occurrence of expression in text,
- `match` – finds all occurrences of expression in text,
- `fullmatch` – matches only if the whole string matches the regular expression,
- `split` – splits into a list based on the splitting expression,
- `escape` – replaces all characters in the pattern.

Regular expressions are used within many methods that we go through in the next slides.

Word and sentence comparison methods

String comparison methods available in Python:

- Levenshtein distance,
- Damerau-Levenshtein distance,
- Jaro distance,
- Jaro-Winkler distance,
- Match rating approach comparison,
- Hamming distance,
- Gestalt pattern matching.

You can use at least two libraries:

- DiffliB – <https://docs.python.org/3.6/library/difflib.html>,
- Jellyfish – <https://pypi.org/project/jellyfish/>.

String comparison – Levenshtein Distance

The Levenshtein distance is a number of insertion, deletion or replacement changes that needs to be done to get the same strings.

It is a number that is equal or higher than 0. It can be normalized to get a number from 0 to 1.

compared words	word length
training	8
trains	6

The distance for both words is 3. After the normalization the distance is $\frac{3}{8} = 0.375$.

String comparison – Gestalt pattern matching

This solution can be formulated as:

$$G_{PM} = \frac{\text{\#same characters}}{\text{\#total characters}}.$$

For the same example we have 5 same characters in each word and four that are different. This makes the G_{PM} value:

$$G_{PM} = \frac{10}{14} = 0.7142.$$

SQL Like vs. Full-text search

The full-text search is in most cases much faster than a Like query.

$$\text{bm25}(D, Q) = -1 \sum_{i=1}^n \text{IDF}(q_i) \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \frac{|D|}{\text{avg}|})},$$

where:

- $|D|$ is the number of tokens in the current document,
- k_1 and b are constants with values 1.2 and 0.75,
- $\text{avg}|$ is the average number of tokens.

SQL Like vs. Full-text search

IDF is the inverse-document-frequency of query phrase i and is formulated as:

$$\text{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5},$$

where:

- N is the total number of rows in table,
- $n(q_i)$ is the total number of rows that contain at least one instance of phrase i .

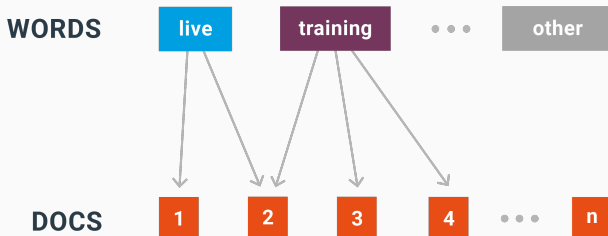
$f(q_i, D)$ is the phrase frequency of phrase i :

$$f(q_i, D) = \sum_1^{nc} w_c \cdot n(q_i, c),$$

where:

- w_c are the weights assigned to columns,
- $n(q_i, c)$ is the number of occurrences of phrase i in column c of the current row.

This is a **live** training.



Natural Language Processing

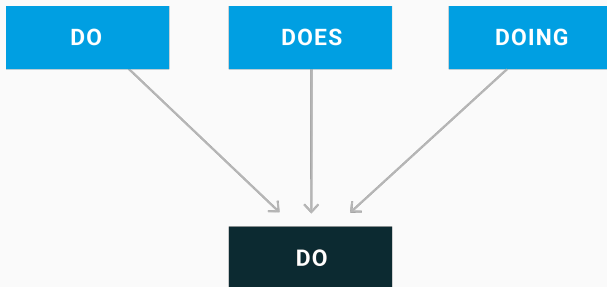
NLP methods used for sentence comparison

There are three popular methods that are used in NLP for text processing:

- tokenization,
- lemmatization,
- stemming.

Tokenization divides a sentence into separate words.

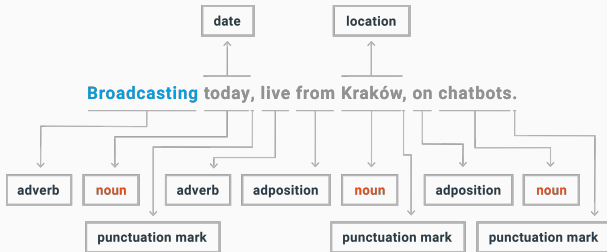
Lemmatization and stemming



There are three popular NLP methods that make it easier to understand written text:

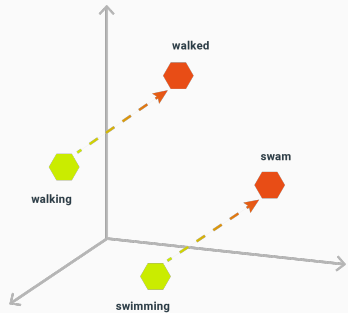
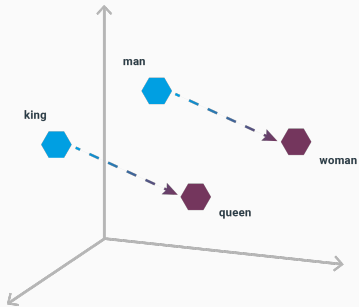
- part of speech,
- noun chunk,
- named entity recognition.

More advanced NLP methods

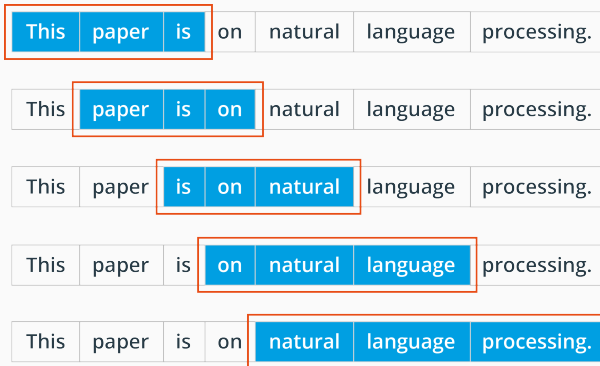


Text understanding

Word vectorization



Word vectorization – concept



Word vectorization – methods

The most popular methods that are used to create a space of vectorized words are:

- bag of words,
- tf-idf,
- transfer learning,
- n-gram model,
- skip-thought vectors.

BAG OF WORDS

This is a paper on machine learning. Deep learning is a recent trend.



{ This, is, a, paper, on, machine, learning, deep, recent, trend }

1 [1 1 1 1 1 1 1 0 0 0]

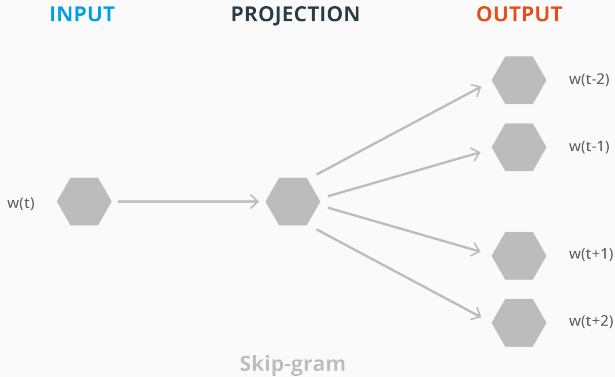
2 [0 1 1 0 0 0 1 1 1 1]

Distance metrics

Also known as similarity or dissimilarity measures.

Measure name	equation
Manhattan distance	$\rho_{Man}(x_r, x_s) = \sum_{i=1}^n x_{ri} - x_{si} \quad (1)$
Chebyshev distance	$\rho_{Ch}(x_r, x_s) = \max_{1 \leq i \leq n} x_{ri} - x_{si} \quad (2)$
Frecht distance	$\rho(x_r, x_s) = \sum_{i=1}^d \frac{ x_{ri} - x_{si} }{1 + x_{ri} + x_{si} } \frac{1}{2^i} \quad (3)$
Canberra distance	$\rho(x_r, x_s) = \sum_{i=1}^d \frac{ x_{ri} - x_{si} }{ x_{ri} + x_{si} } \quad (4)$
Post office distance	$\rho_{pos}(x_r, x_s) = \begin{cases} \rho_{Min}(x_r, 0) + \rho_{Min}(0, x_s), & \text{for } x_r \neq x_s, \\ 0, & \text{for } x_r = x_s \end{cases} \quad (5)$
Bray-Curtis distance	$\rho_{bc}(x_r, x_s) = \frac{\sum_{i=1}^d x_{ri} - x_{si} }{\sum_{i=1}^d (x_{ri} + x_{si})} \quad (6)$

Word vectorization – concept



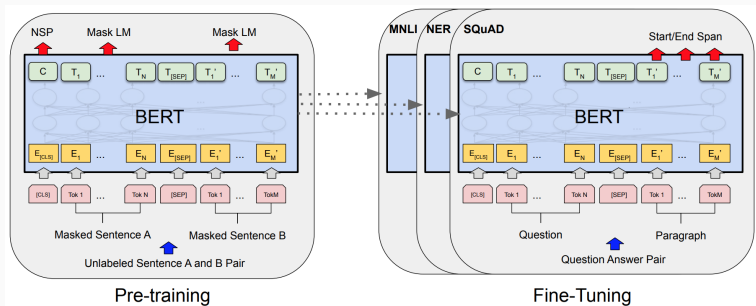
Modern word vectorization methods

The BoW or Tf-idf methods are simple, but currently more advanced methods are used like:

- Word2Vec – <https://arxiv.org/abs/1301.3781>,
- GloVe – <https://nlp.stanford.edu/projects/glove/>,
- BERT (VisualBERT) – <https://arxiv.org/abs/1810.04805>.

BERT

Source: <https://arxiv.org/abs/1810.04805>



Chatbots

Chatbots – a new interface

Bots are a new way of communication between the user and the app ¹.



¹Designing Bots, 1st Edition. *Amir Shevat*, O'Reilly Media 2017

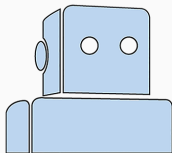
Bots can be divided into a few types, based on:

- interface – automation, audio or text,
- privacy – on-site and online,
- usage – superbots, domain-driven, etc.²

¹Designing Bots, 1st Edition. *Amir Shevat*, O'Reilly Media 2017



{LawGeex}



You can find a short explanation on how to start in the chatbots notebooks:

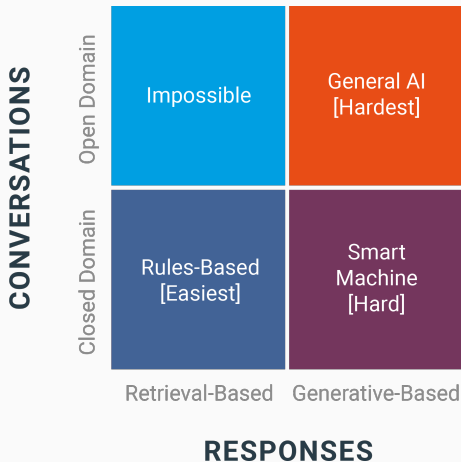
https://github.com/codete/oreilly-intelligent-bots/blob/master/Chatbot_Integrations.ipynb



*Are chatbots
intelligent like
humans?*



Bot matrix



¹Ultimate Guide to Leveraging NLP and Machine Learning for your Chatbot. Stefan Kojouharov, Chatbots Life 2016

Phrases list

Show status of recruitment.

What is the weather in Berlin?

Hi!

Hire candidate <name>.

Answers list

We have currently X candidates.

It is <current weather>.

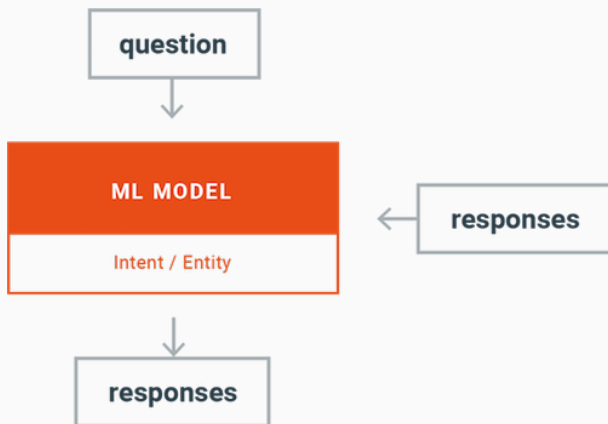
Hi. How are you?

Sent an email to <email>.

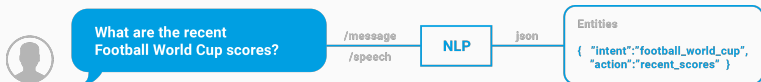


No valid phrase found

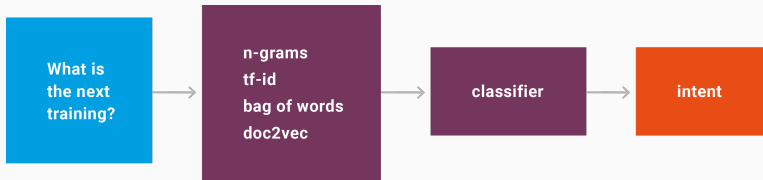
Retrieval-based – basics



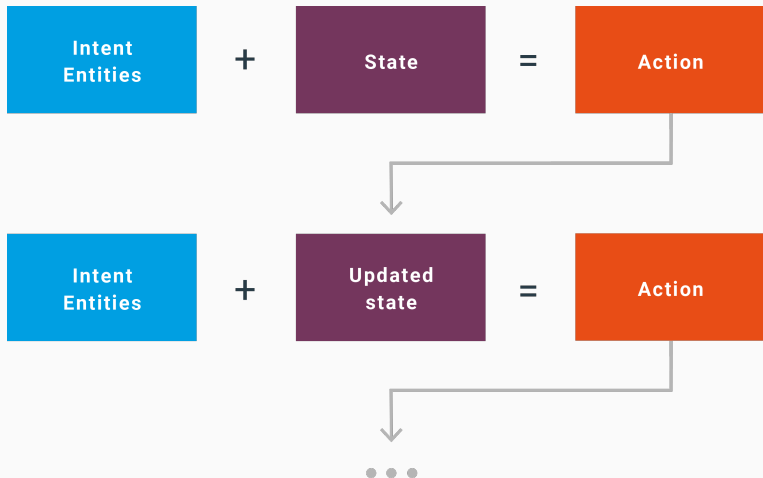
Entities and intents



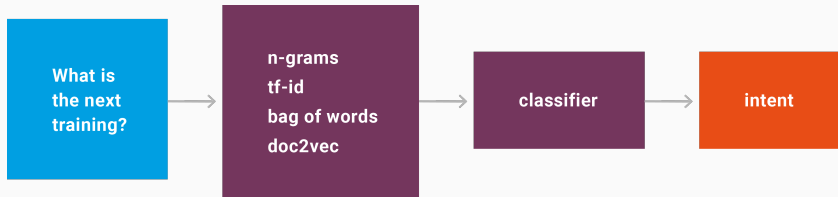
Entities and intents



Rasa NLU engine



Rasa intent learning process



Advantages

Rule-based chatbots:

- predictable,
- clear principles,
- cheap.

Generative-based chatbots:

- generic, intelligent answers,
- raw data as training data set.

Retrieval-based chatbots:

- identify the intent,
- usually easy to train,
- do not need too many questions/answers,
- more intelligent than rule-based.

Rule-based chatbots:

- too simple for most cases,
- not really intelligent.

Retrieval-based chatbots:

- limited to questions/answers
- not a generic solution.

Generative-based chatbots:

- usually take longer to train,
- needs a dataset, usually a huge one,
- sometimes unpredictable.

Text generation

Natural Language Generation is a part of Natural Language Processing. The goal of NLG is to generate a sentence or the whole document that has a logical sense, follows the grammar and answers the question properly if we deal with a bot.

There are plenty of methods that can be used for text generation. The most popular are:

- n-gram model,
- recurrent neural network,
- autoencoders,
- generative adversarial network.

N-gram model

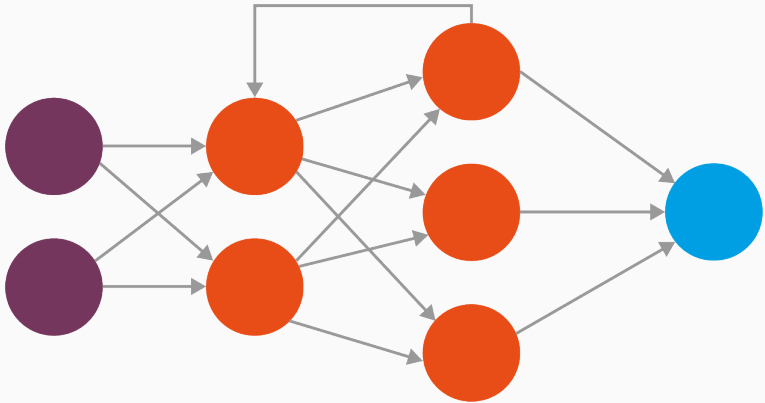
This	paper	is	on	natural	language	processing.
------	-------	----	----	---------	----------	-------------

This	paper	is	on	natural	language	processing.
------	-------	----	----	---------	----------	-------------

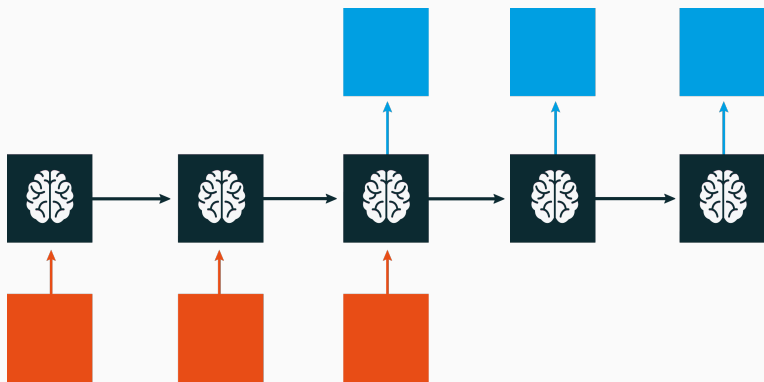
This	paper	is	on	natural	language	processing.
------	-------	----	----	---------	----------	-------------

This	paper	is	on	natural	language	processing.
------	-------	----	----	---------	----------	-------------

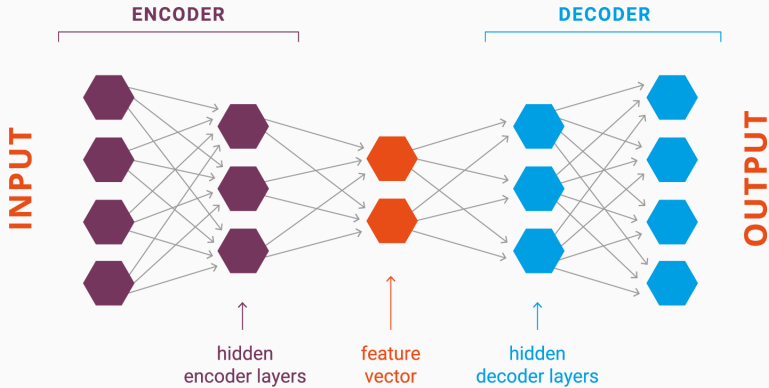
This	paper	is	on	natural	language	processing.
------	-------	----	----	---------	----------	-------------



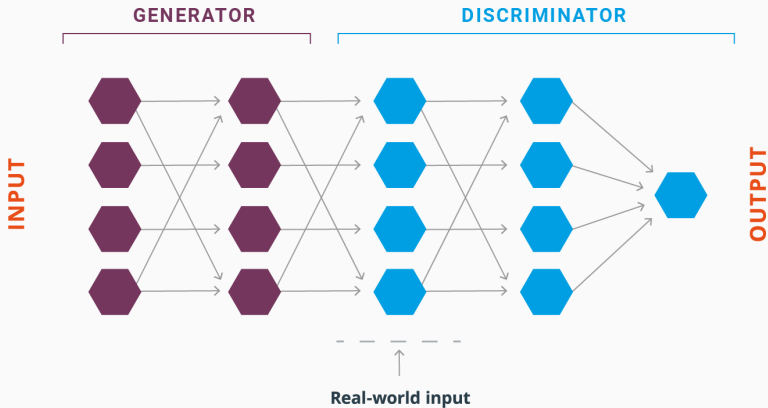
LSTM



Autoencoders



Generative Adversarial Networks



Working examples

There are many open source trained models available. Here are a few worth mentioning:

- chatterbot – a chatbot implementation
<http://chatterbot.readthedocs.io/>
- DeepQA – uses RNN and has a web interface
<https://github.com/Conchylicultor/DeepQA>
- Generative Conversational Agents – uses LSTM, RNN and GAN
[https://github.com/oswaldoludwig/Adversarial-Learning-for-Generative-Conversational-Agents.](https://github.com/oswaldoludwig/Adversarial-Learning-for-Generative-Conversational-Agents)

Research datasets

A few datasets useful for your research:

- **SQuAD** – reading comprehension dataset, consists of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to each question is a segment of text,
<https://rajpurkar.github.io/SQuAD-explorer/>,
- **Cornell Movie Dialogs Corpus** – movie dialogs,
https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html,
- **DeepMind datasets** – AQua is a dataset of questions and answers,
<https://github.com/deepmind/AQua>, more datasets from DeepMind: <https://deepmind.com/research/open-source/open-source-datasets/>,
- **DMQA** – Daily Mail and CNN articles data sets,
<https://cs.nyu.edu/~kcho/DMQA/>,
- **MS MARCO** – Microsoft MACHine Reading COmprehension Dataset, <http://www.msmarco.org/dataset.aspx>.

Quality metrics

Quality metrics for NLP

In most cases we use the same quality metrics as in regular machine learning models, if any machine learning model is used for NLP.

Other methods that can be used for NLP methods validation, especially, text generation:

- spell check,
- grammar check,
- typo check.

Questions?