Important website:

<http://nlpprogress.com/english/simplification.html>

<https://guillaumegenthial.github.io/>

Code for **access (controllable sentence simplification)**：

<https://github.com/facebookresearch/access>

Code for Neural text simplification:

<https://github.com/senisioi/NeuralTextSimplification>

Paradigm for QA:

Information retreiveal based and knowledge-based graph.

Normalization:

Bert cannot deal with misspelling (beginning vs begining ), contraction like cant

well-known off-the-shelf SMT(statistical machine translation) tools like Moses

Neural machine translation

<https://towardsdatascience.com/creating-word-embeddings-for-out-of-vocabulary-oov-words-such-as-singlish-3fe33083d466>

**Text summarization. (deep learning, bert abstract , extractive(NLK lilbrary. ))**

<https://www.youtube.com/watch?v=_d0OXm0dRZ4>

abstractive summarization:

<https://towardsdatascience.com/summarization-has-gotten-commoditized-thanks-to-bert-9bb73f2d6922>

<https://arxiv.org/pdf/1908.08345.pdf>

supervised learning to build summary models . <https://www.analyticsvidhya.com/blog/2019/06/comprehensive-guide-text-summarization-using-deep-learning-python/>

extractive:

unsupervised learning:

summaries by choosing a subset of the sentences in the original text.

This website shows how to select top 5 sentences in the article using page-rank algorithm.

The link between each sentence is the cos similarity between sentences.

U find the eigenvector, then u can select the top sentences.

<https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/>

Knowledge graph:

Wiki data using Wiki data API

**Relation extraction:**

from an existing Knowledge Base (KB), such as Wikipedia, DBpedia, Wikidata, Freebase, Yago.

Semi-supervised:

Unsupervised:

If the relation is very common in the web, then it is indeed a relation.

For testing , we just draw random samples from the web and then decide manually the accuracy rate.

[OpenIE 5.0](https://github.com/dair-iitd/OpenIE-standalone) and [Stanford OpenIE](https://nlp.stanford.edu/software/openie.html) are two open-source systems that does this. They are more modern than TextRunner (which was just used here to demonstrate the paradigm).

Graph representation to vector representation:

A close up of a map

Description automatically generated

Using random walk to find similarity of nodes,

The dot product of vector representation of node is the probability that a and b co-occur on a random walk over the network.

Thus the information of connection of node is embedded in the spacial representation(the angle)

It is a bit like calculating word embedding for word and text.

<https://www.analyticsvidhya.com/blog/2019/11/graph-feature-extraction-deepwalk/>

Knowledge representation learning:

<https://github.com/thunlp/TensorFlow-TransX>