Question Answering and Chatbots

2nd Practical exercise – Question Classification

Aleksandr Perevalov

aleksandr.perevalov@hs-anhalt.de

October 20, 2021

Hochschule Anhalt

Anhalt University of Applied Sciences

Text Classification Task

Text Classification Task

Having **a sequence of text documents** $D = [d_1, d_2, ...d_i]$ each document d_i has a **class** c_i assigned to it (**training data**).

Text Classification Task

Having **a sequence of text documents** $D = [d_1, d_2, ...d_i]$ each document d_i has a **class** c_i assigned to it (**training data**).

A classification algorithm has to be trained, such that it can learn to predict document classes based on the given data as precise as possible.

Text Classification Task

Having **a sequence of text documents** $D = [d_1, d_2, ...d_i]$ each document d_i has a **class** c_i assigned to it (**training data**).

A classification algorithm has to be trained, such that it can learn to predict document classes based on the given data as precise as possible.

The classifier has to predict classes not only for the training data but for previously unseen data.

Text Classification Task

Having **a sequence of text documents** $D = [d_1, d_2, ...d_i]$ each document d_i has a **class** c_i assigned to it (**training data**).

A classification algorithm has to be trained, such that it can learn to predict document classes based on the given data as precise as possible.

The classifier has to predict classes not only for the training data but for previously unseen data.

Types of classification:

Text Classification Task

Having **a sequence of text documents** $D = [d_1, d_2, ...d_i]$ each document d_i has a **class** c_i assigned to it (**training data**).

A classification algorithm has to be trained, such that it can learn to predict document classes based on the given data as precise as possible.

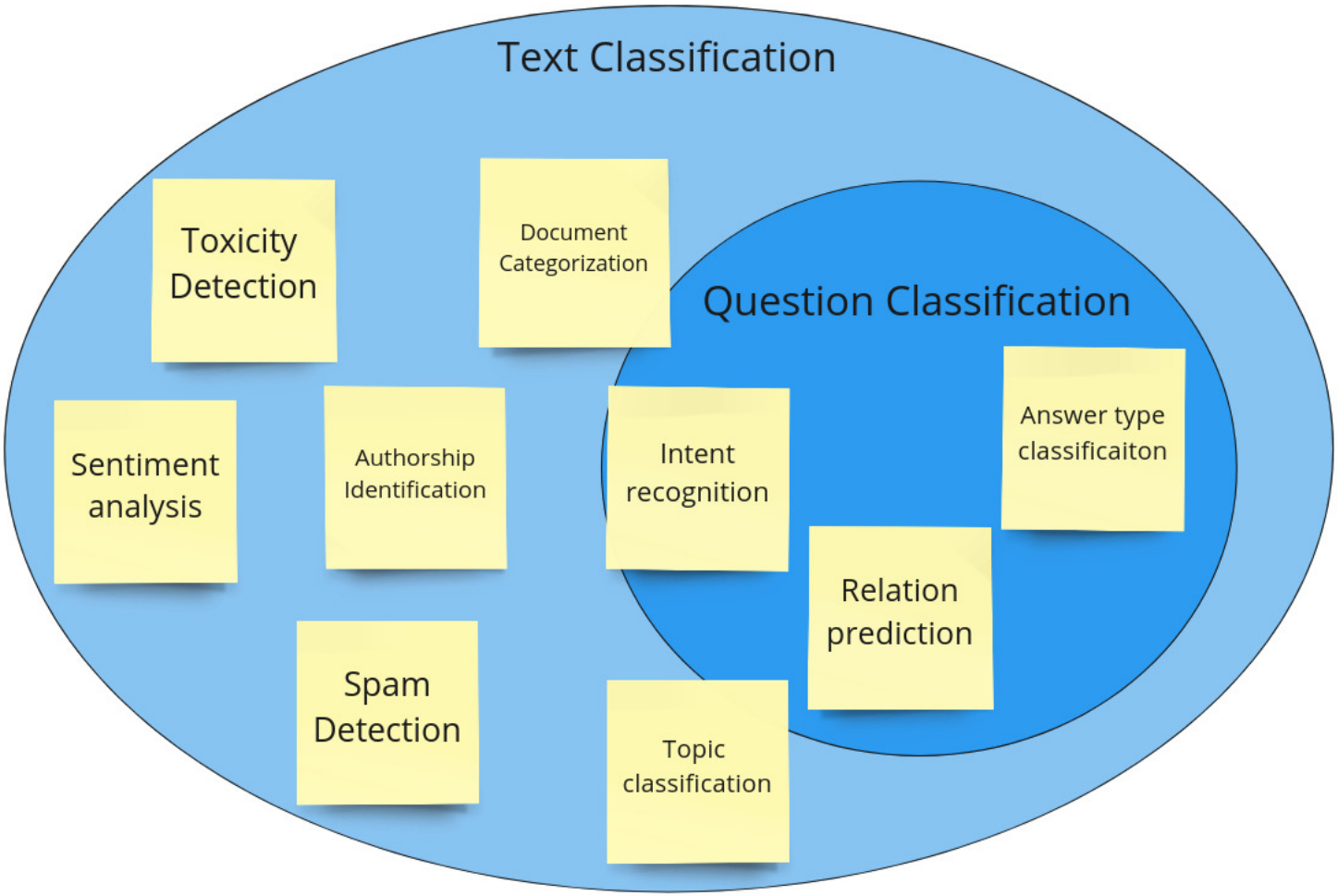
The classifier has to predict classes not only for the training data but for previously unseen data.

Types of classification:

Binary	2 classes
Multi-class	$>$ 2 classes (typically)
Multi-label	a data item might have $\geq$ 1 class

Text or Question classification

Text or Question classification



Question classification in QA

Question classification in QA

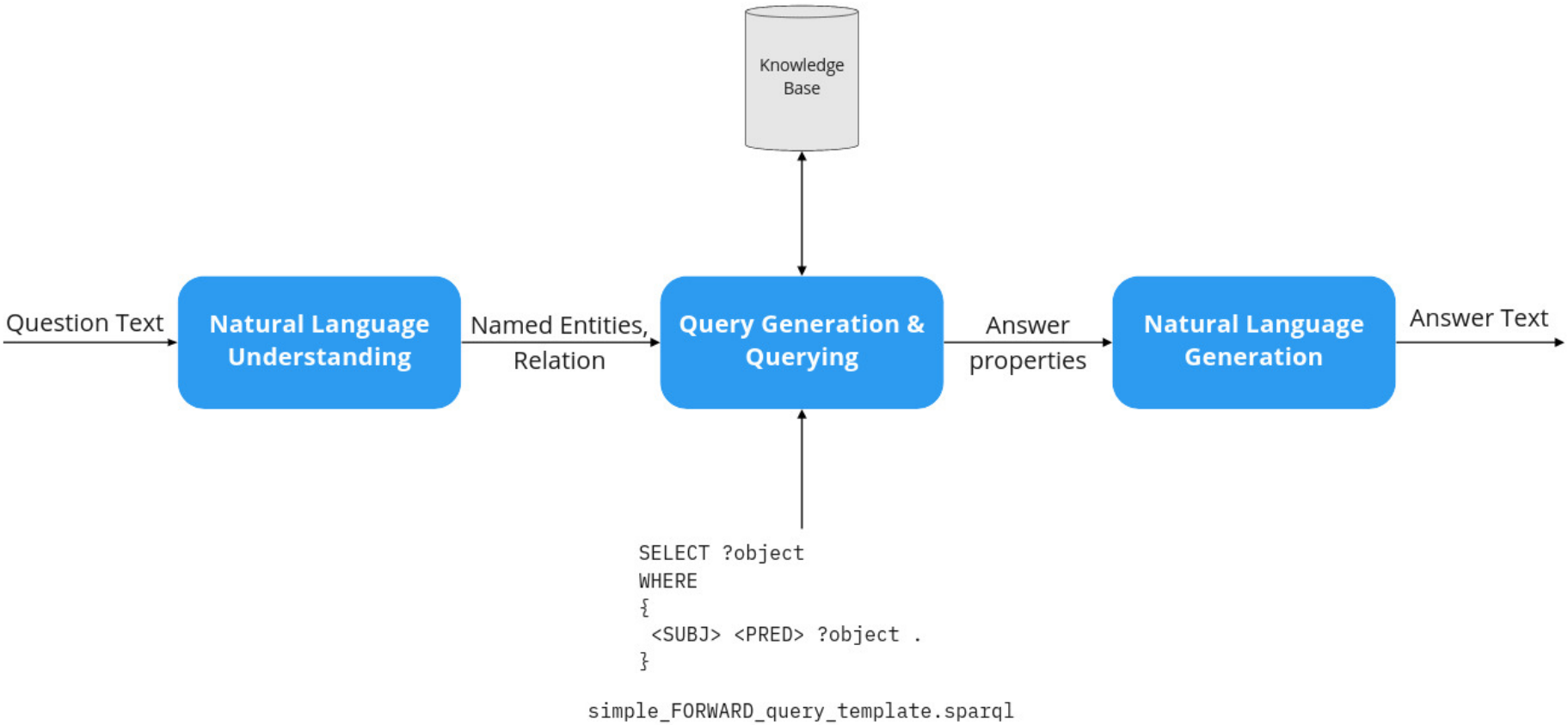
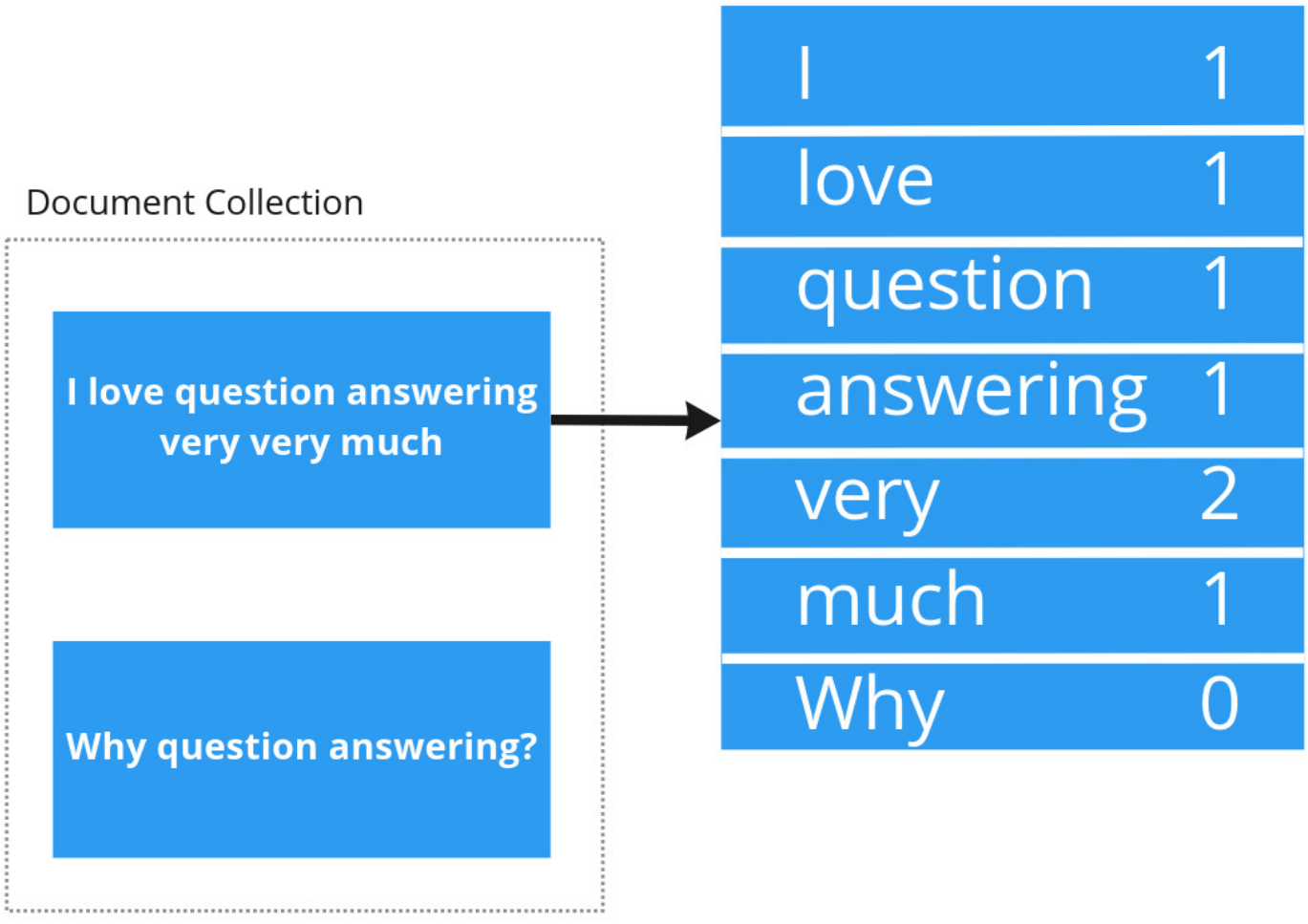


Figure: QA system architecture for "Simple Questions"

Text-to-vector transformation

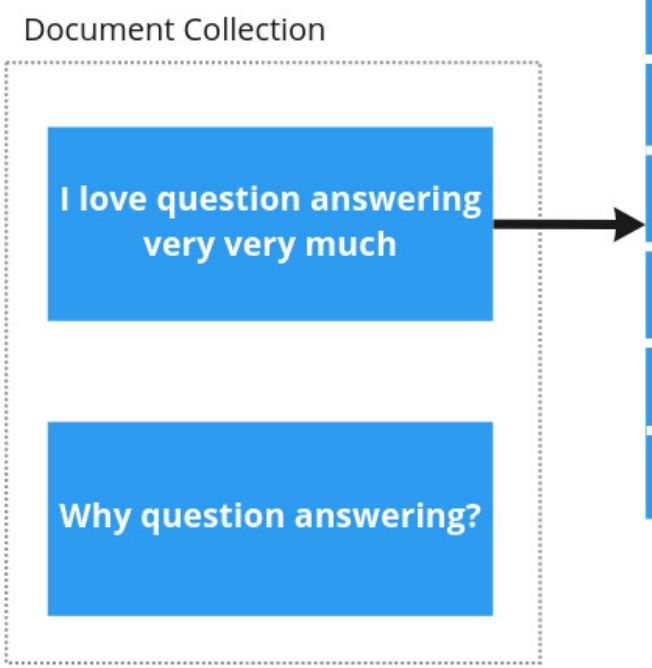
Text-to-vector transformation

Bag of Words



Text-to-vector transformation

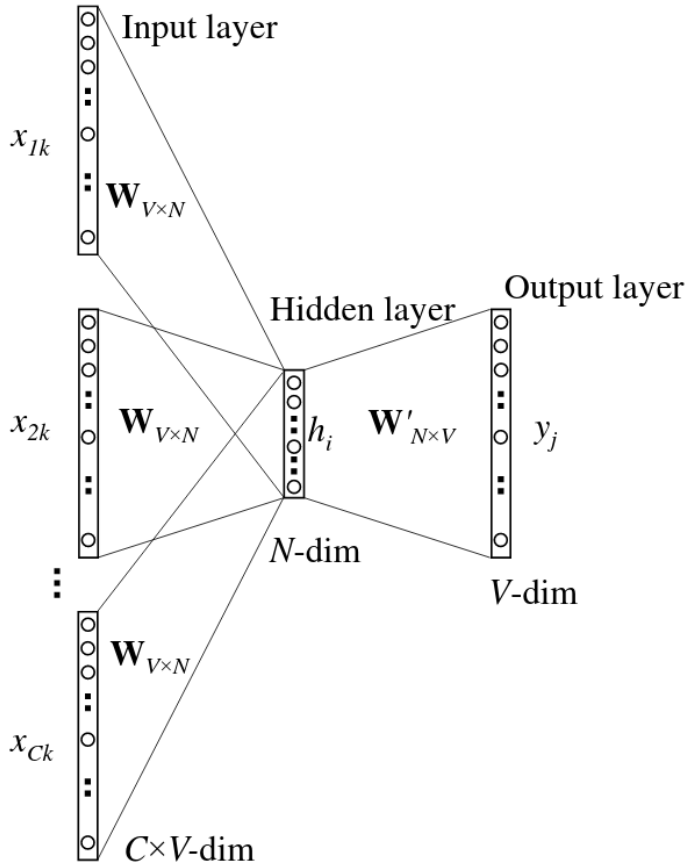
Term Frequency - Inverse Document Frequency (TF-IDF)



	TF	IDF	TFxIDF
I	1/7	log(2/1)	0.099
love	1/7	log(2/1)	0.099
question	1/7	log(2/2)	0.0
answering	1/7	log(2/2)	0.0
very	2/7	log(2/1)	0.198
much	1/7	log(2/1)	0.099
Why	0/7	log(2/1)	0.0

Text-to-vector transformation

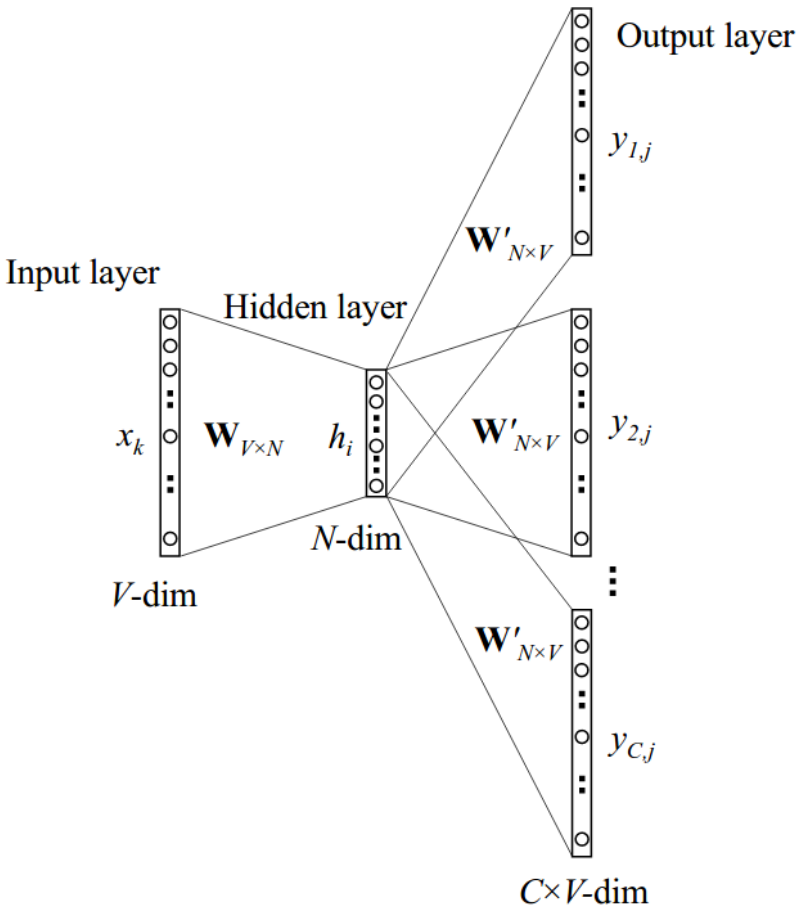
Word2Vec (2013) – Continious Bag of Words [1]



[1] `https://arxiv.org/pdf/1411.2738.pdf`

Text-to-vector transformation

Word2Vec (2013) – Skip-Gram [1]



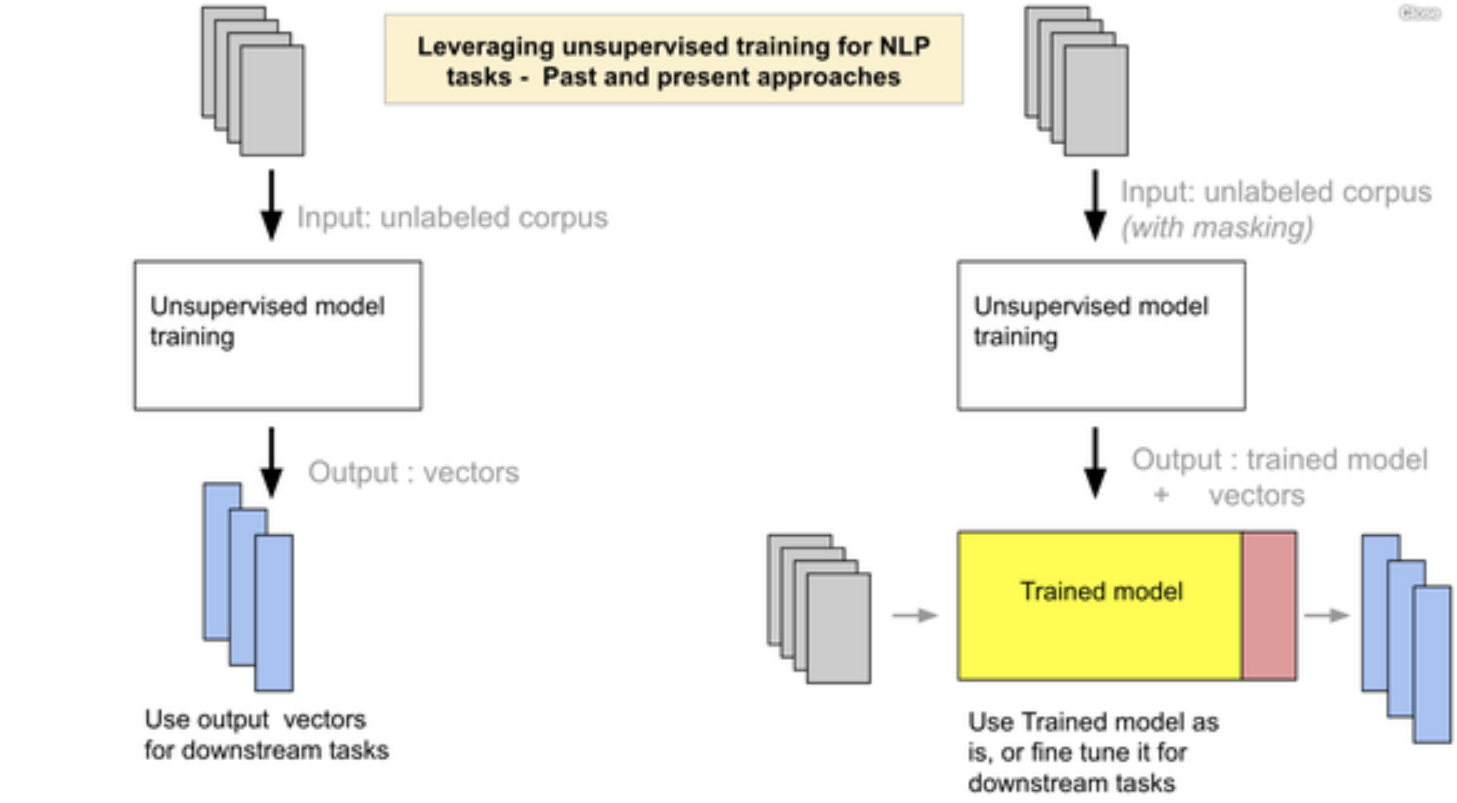
[1] https://arxiv.org/pdf/1411.2738.pdf

Text-to-vector transformation

fastText (2016) – Same as Word2Vec, but instead of words character n-grams are considered as an input.

For example the word vector “apple” is a sum of the vectors of the n-grams “ap”, “app”, ”appl”, ”apple”, ”apple”, “ppl”, “pple”, ”pple”, “ple”, ”ple”, ”le” (assuming hyperparameters for smallest ngram is 3 and largest ngram is 6).

Modern approaches



[2] https://www.quora.com/What-were-the-most-significant-Natural-Language-Processing-advances-in-2018

Let's do the exercise. Ask me if you have a question.

Exercise 2 – Context

Text classification has many applications in Natural Language Processing. Specifically, in Question Answering & Chatbots, it can be used as a **Relation (Predicate) Prediction** component.

Exercise 2 – Context

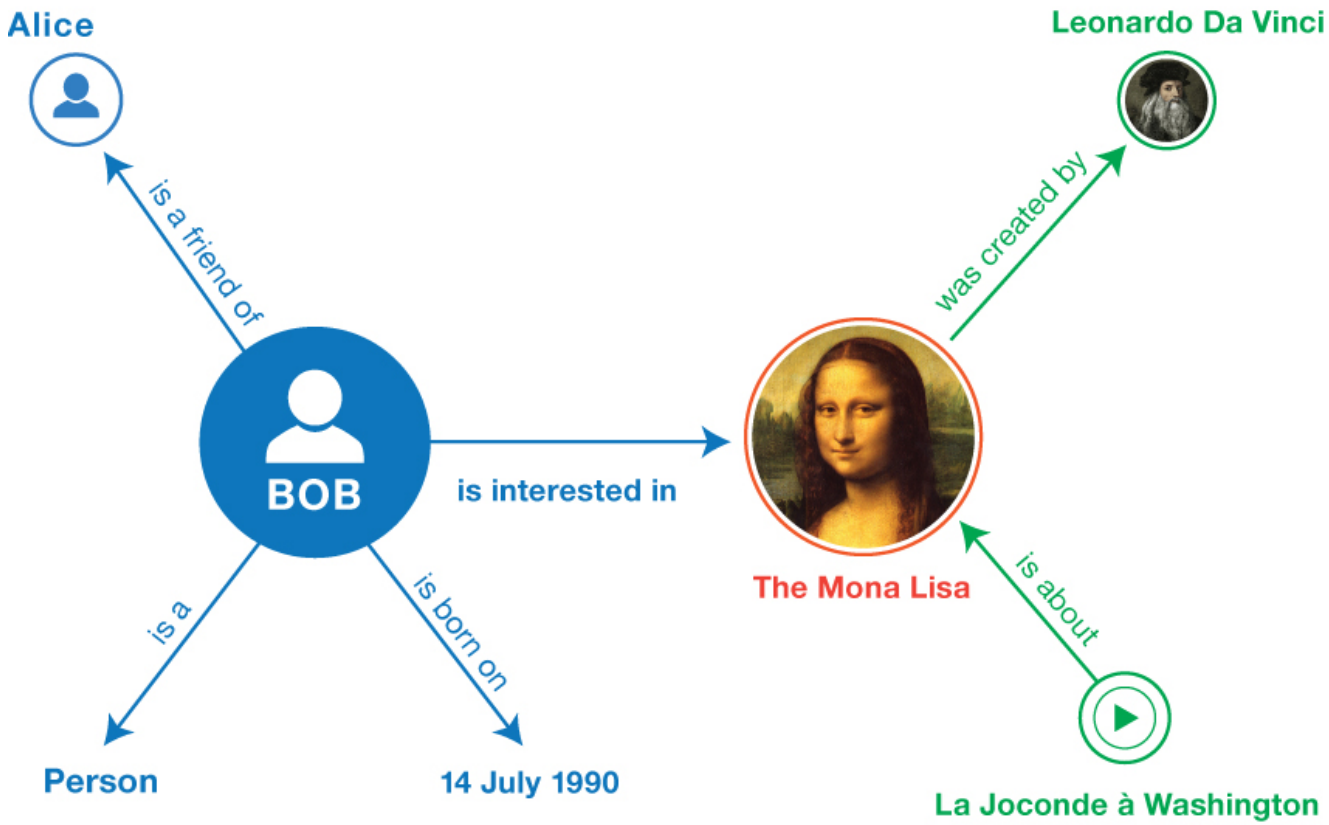
Relation (or Predicate) in terms of knowledge graphs is an **edge that is connecting two nodes (or entities)**. For example, having a triple: <Mona_Lisa> <?> <Leonardo_da_Vinci> the relation <?> is <Author> (or e.g. <Was_Created_By>).

Exercise 2 – Context

In this regard, Relation Prediction is the task of **recognizing a relation, based on a textual question**. In this case, question: "Who is the author of Mona Lisa?" has relation "Author" (or e.g. "Was created by").

Exercise 2 – Context

In this regard, Relation Prediction is the task of **recognizing a relation, based on a textual question**. In this case, question: "Who is the author of Mona Lisa?" has relation "Author" (or e.g. "Was created by").



Exercise 2 - Part 1

Depending on your exercise variant analyze data in `train.csv` and `test.csv` by calculating:

Exercise 2 - Part 1

Depending on your exercise variant analyze data in `train.csv` and `test.csv` by calculating:

- Number of questions per class;

Exercise 2 - Part 1

Depending on your exercise variant analyze data in `train.csv` and `test.csv` by calculating:

- Number of questions per class;
- Average token number in a question;

Exercise 2 - Part 1

Depending on your exercise variant analyze data in `train.csv` and `test.csv` by calculating:

- Number of questions per class;
- Average token number in a question;
- Average character number a question.

Exercise 2 - Part 1

Depending on your exercise variant analyze data in `train.csv` and `test.csv` by calculating:

- Number of questions per class;
- Average token number in a question;
- Average character number a question.

Implement a question classification algorithm based on provided training data (see variant). You can use a Rule-Based approach (e.g., keyword classifier) or a Machine Learning approach (e.g., Bag of Words + Logistic Regression).

Exercise 2 - Part 1

Depending on your exercise variant analyze data in `train.csv` and `test.csv` by calculating:

- Number of questions per class;
- Average token number in a question;
- Average character number a question.

Implement a question classification algorithm based on provided training data (see variant). You can use a Rule-Based approach (e.g., keyword classifier) or a Machine Learning approach (e.g., Bag of Words + Logistic Regression).
For evaluation use the following metrics: Precision, Recall, F1 Score.
See details in Moodle/Github.

Exercise 2 - Part 2

Now when you have your classification algorithm, you are asked to integrate it inside a Web service, that satisfies the following:

Exercise 2 - Part 2

Now when you have your classification algorithm, you are asked to integrate it inside a Web service, that satisfies the following:

- Request path name: /predict;

Exercise 2 - Part 2

Now when you have your classification algorithm, you are asked to integrate it inside a Web service, that satisfies the following:

- Request path name: /predict;
- Request type: POST;

Exercise 2 - Part 2

Now when you have your classification algorithm, you are asked to integrate it inside a Web service, that satisfies the following:

- Request path name: /predict;
- Request type: POST;
- Input structure of request: `{'question': 'Question To Classify'}`

Exercise 2 - Part 2

Now when you have your classification algorithm, you are asked to integrate it inside a Web service, that satisfies the following:

- Request path name: /predict;
- Request type: POST;
- Input structure of request: {'question': 'Question To Classify'}
- Output structure of request: {'predicted_relation': 'RELATION'}

Exercise 2 - Part 2

Now when you have your classification algorithm, you are asked to integrate it inside a Web service, that satisfies the following:

- Request path name: /predict;
- Request type: POST;
- Input structure of request: {'question': 'Question To Classify'}
- Output structure of request: {'predicted_relation': 'RELATION'}

After you establish a Web service, run all your questions through the /predict method and write the predictions into a structured JSON file.

Plan for the Exercise 3: SPARQL Queries

SPARQL – query language for knowledge graphs stored in RDF.

Plan for the Exercise 3: SPARQL Queries

SPARQL – query language for knowledge graphs stored in RDF.

- Learn syntax;

Plan for the Exercise 3: SPARQL Queries

SPARQL – query language for knowledge graphs stored in RDF.

- Learn syntax;
- Given a set of questions – write SPARQL for DBpedia;

Plan for the Exercise 3: SPARQL Queries

SPARQL – query language for knowledge graphs stored in RDF.

- Learn syntax;
- Given a set of questions – write SPARQL for DBpedia;
- Given a set of questions – write SPARQL for Wikidata;

Plan for the Exercise 3: SPARQL Queries

SPARQL – query language for knowledge graphs stored in RDF.

- Learn syntax;
- Given a set of questions – write SPARQL for DBpedia;
- Given a set of questions – write SPARQL for Wikidata;
- Create a script for executing the queries and writing the output.

0 Introduction;
1 NER & NEL;
2 **Question classification & Web service/API**;
3 SPARQL queries over Knowledge Graphs;
4 Simple KGQA system – based on exercises 0, 1, 2, 3;
5 Qanary Framework – component oriented approach;
6 Simple ODQA system?;
7 Evaluation of QA systems.