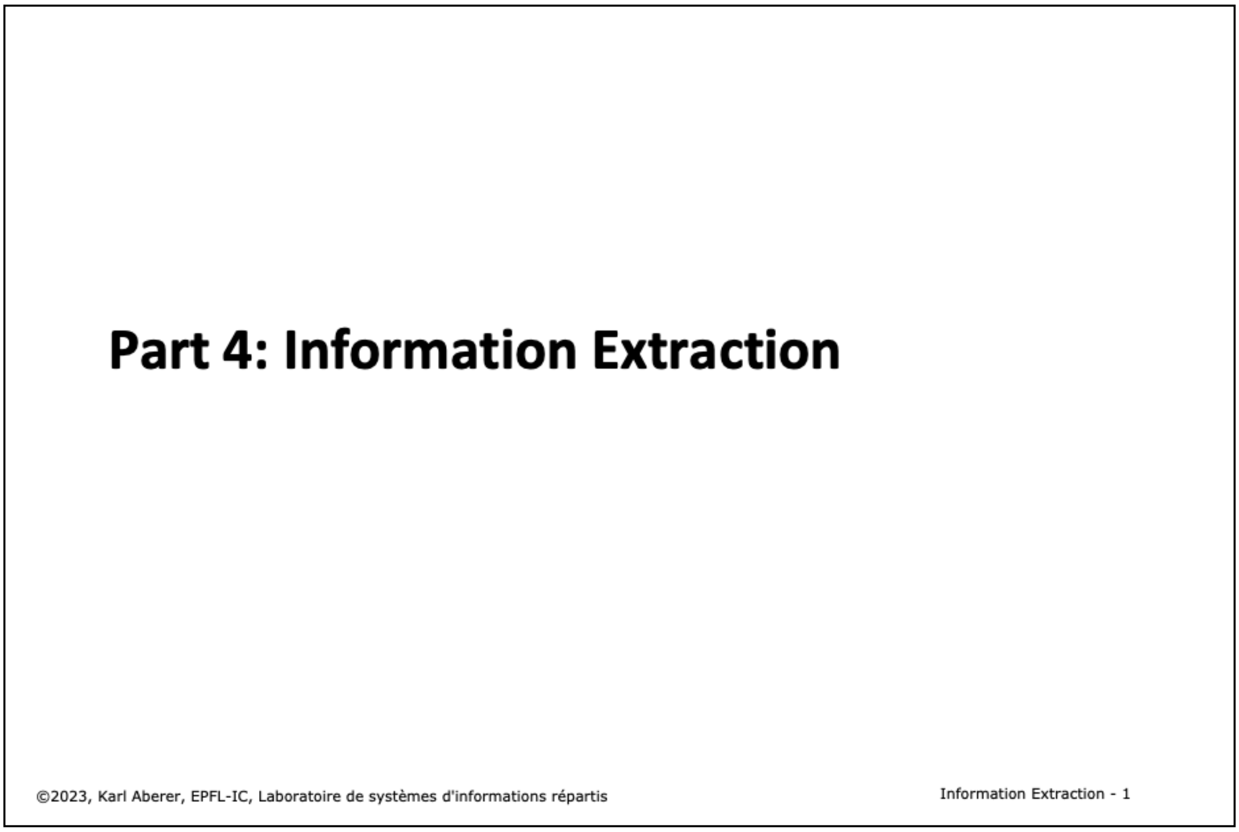
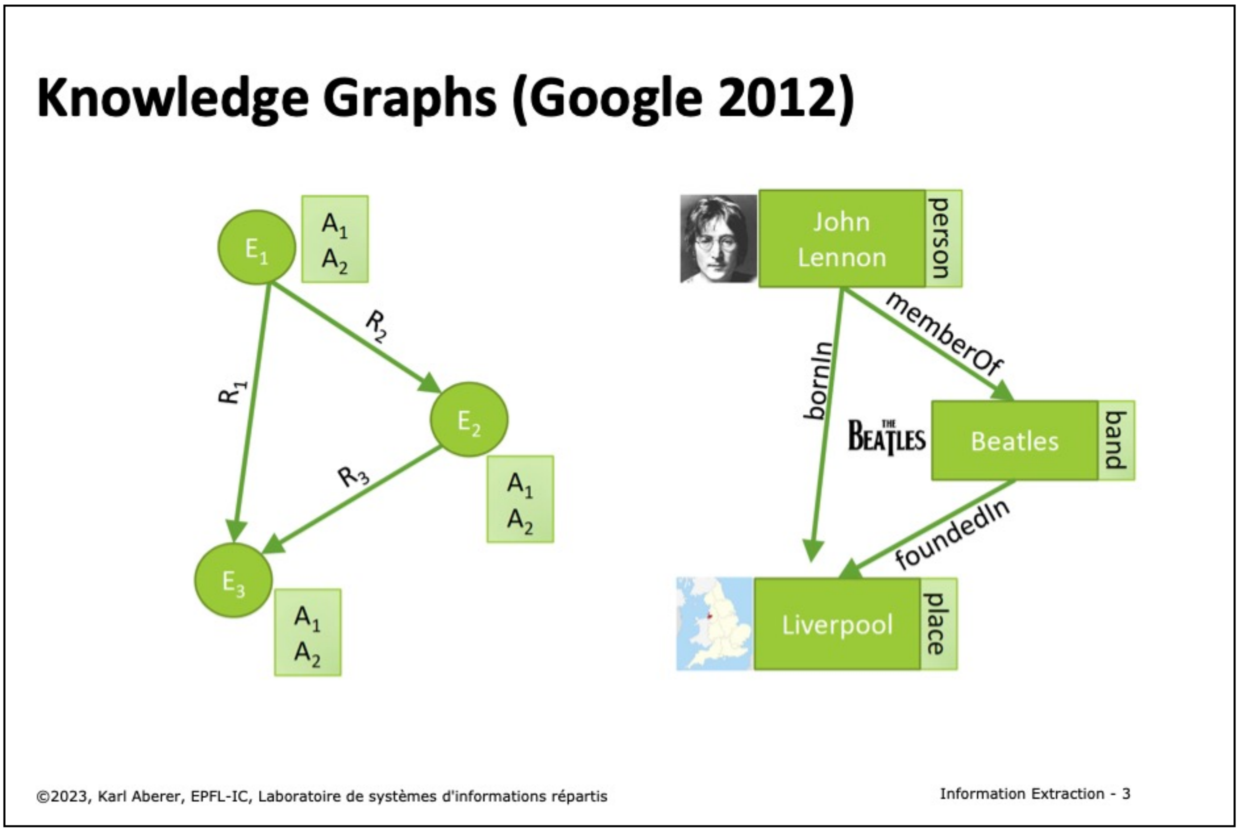


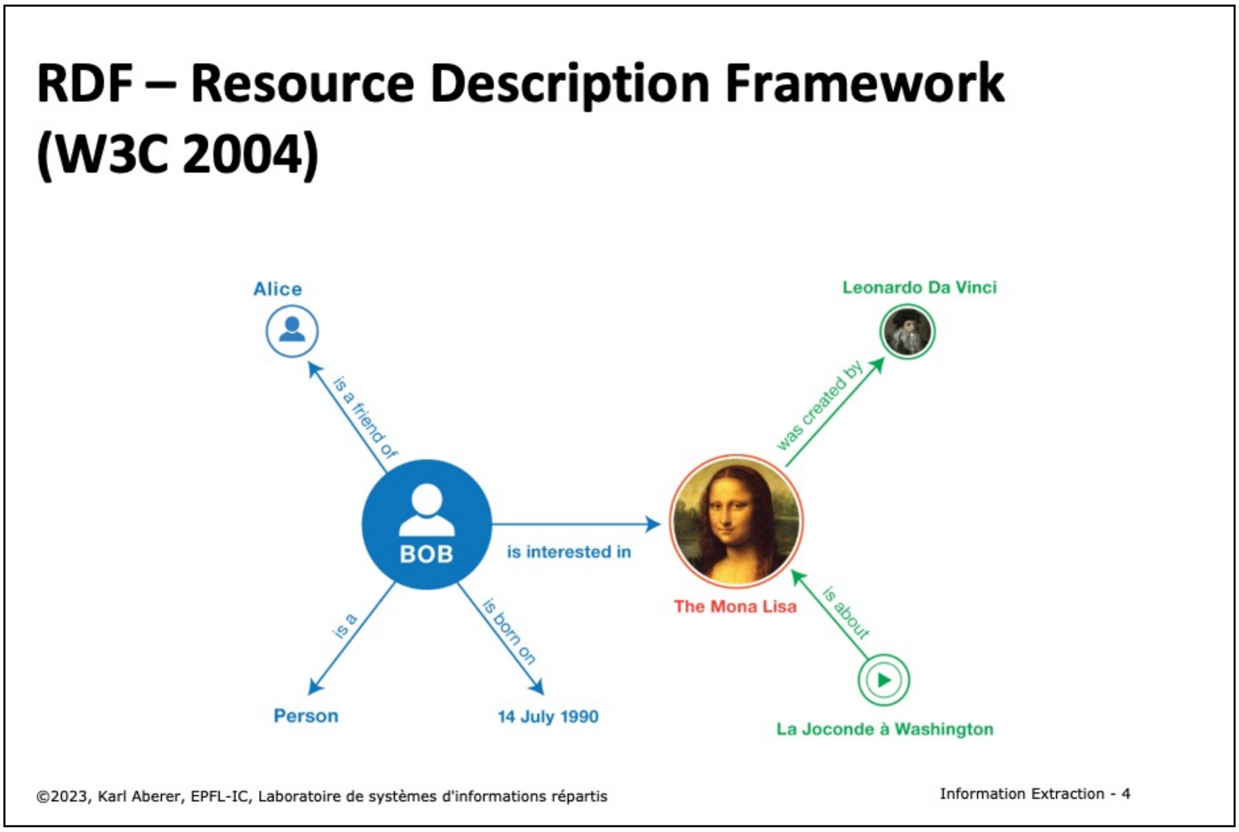
Part 4: Information Extraction

4.1 NAMED ENTITY RECOGNITION



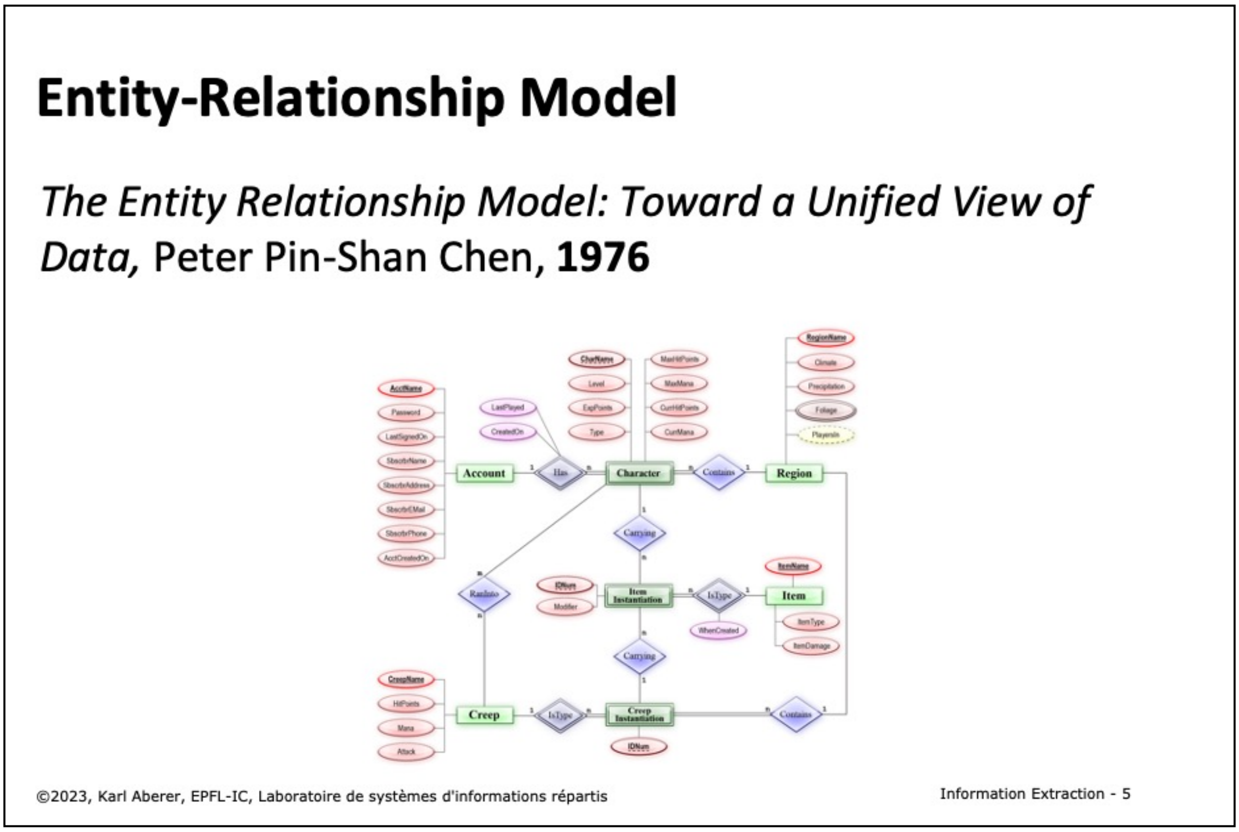
Knowledge Graphs (Google 2012)

One model for representing ontologies that has gained large popularity recently are knowledge graphs. They are based on a graph-based representations of basic first order logic constructs, entities, which are elements from given domains, relationships, which are binary relations and attributes which are relations among entities and values from a standard data structure.



RDF – Resource Description Framework (W3C 2004)

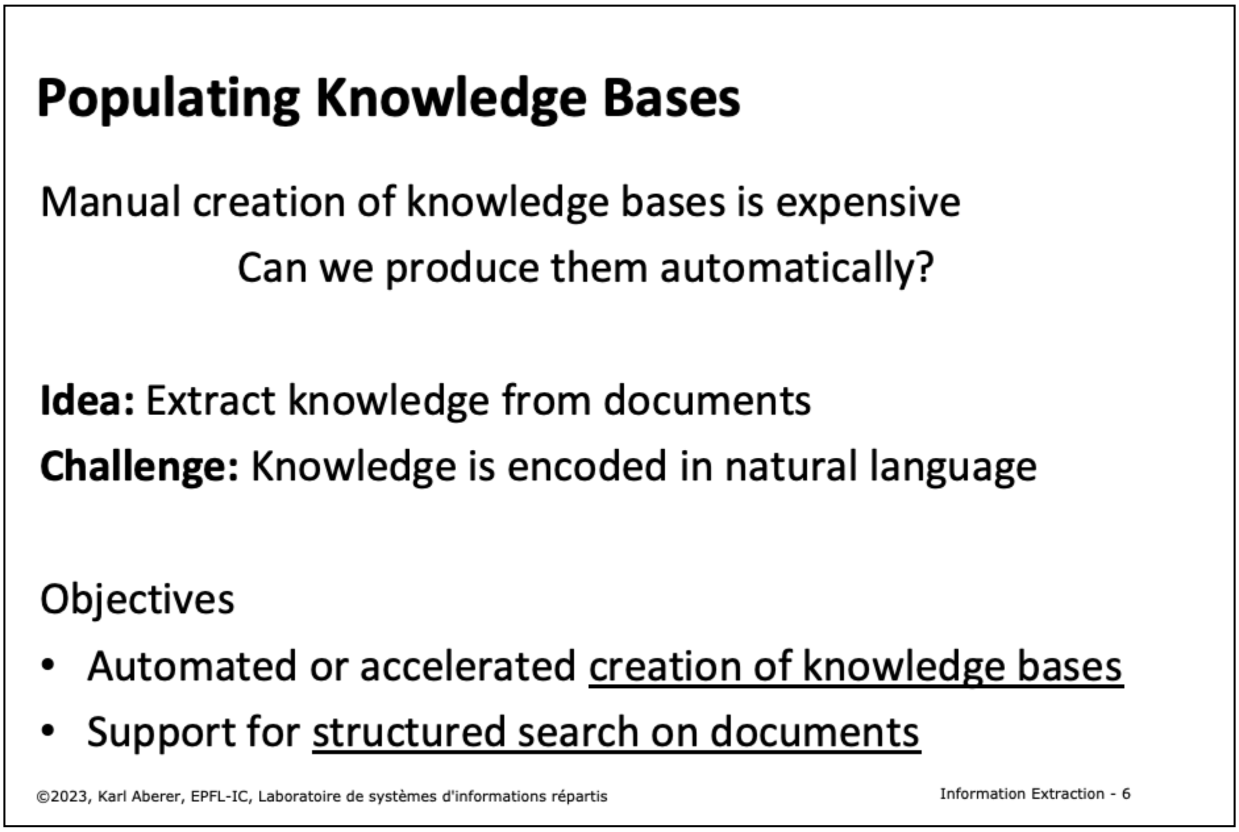
In fact, long before knowledge graph have become popular, a similar model has been developed by the W3C, the Resource Description Framework. It has the same basic concepts as knowledge grpahs, together with some additional modeling primitives. We will provide an overview of RDF in the following.



Entity-Relationship Model

The Entity Relationship Model: Toward a Unified View of Data, Peter Pin-Shan Chen, **1976**

The approach of using concepts from first order logic for modeling reaches however far back in history. In the area of database management systems, the entity-relationship model has been established as a standard approach for data modeling already in 1976. The main difference is in the intended use of the model at the time. It was used to provide a conceptual model of an application domain and then to derive in a mostly automated way a database schema (in other words data structures) in the data model supported by the target database management system. This model typically is the relational data model.



Populating Knowledge Bases

Manual creation of knowledge bases is expensive

Can we produce them automatically?

Idea: Extract knowledge from documents

Challenge: Knowledge is encoded in natural language

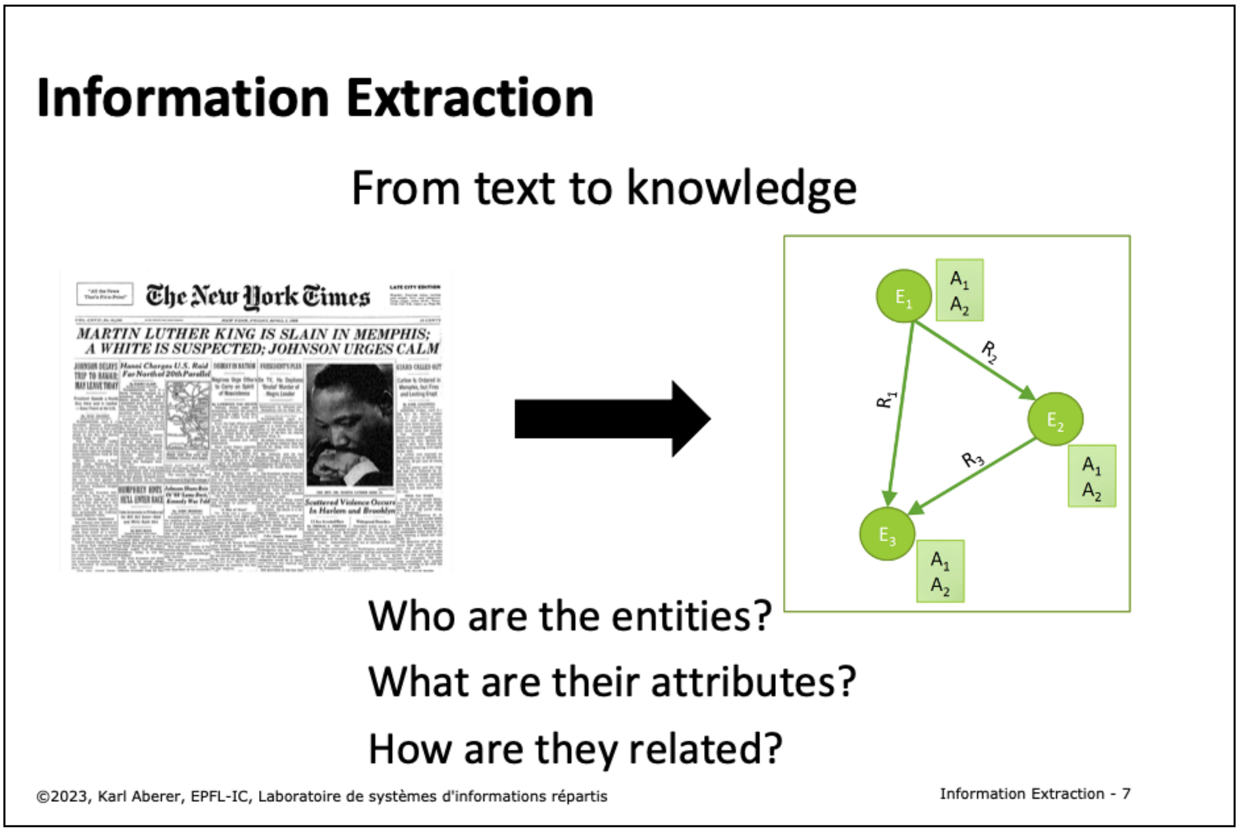
Objectives

- Automated or accelerated creation of knowledge bases
- Support for structured search on documents

Traditionally knowledge bases are created manually, either by experts (e.g., WordNet) or by crowd-sourcing (e.g., WikiData). This is expensive. In the case of WordNet, it took tens of years to construct the knowledge base, in the case of WikiData (resp. WikiPedia) we all know about the notorious difficulty to finance this endeavor. Therefore, an interesting question is whether such knowledge bases could not be automatically constructed.

For automatic construction we can exploit data that is digitally available, e.g., all documents accessible on the Web. These documents encode massive human knowledge in natural language. The challenge is to extract such knowledge by analyzing natural language text, which is not an easy problem.

The results would, however, be immensely useful. First, we could create massive knowledge bases in a nearly automated way, and furthermore these knowledge bases could be used to annotate documents, for supporting more expressive and precise searches and analyses.



Information Extraction

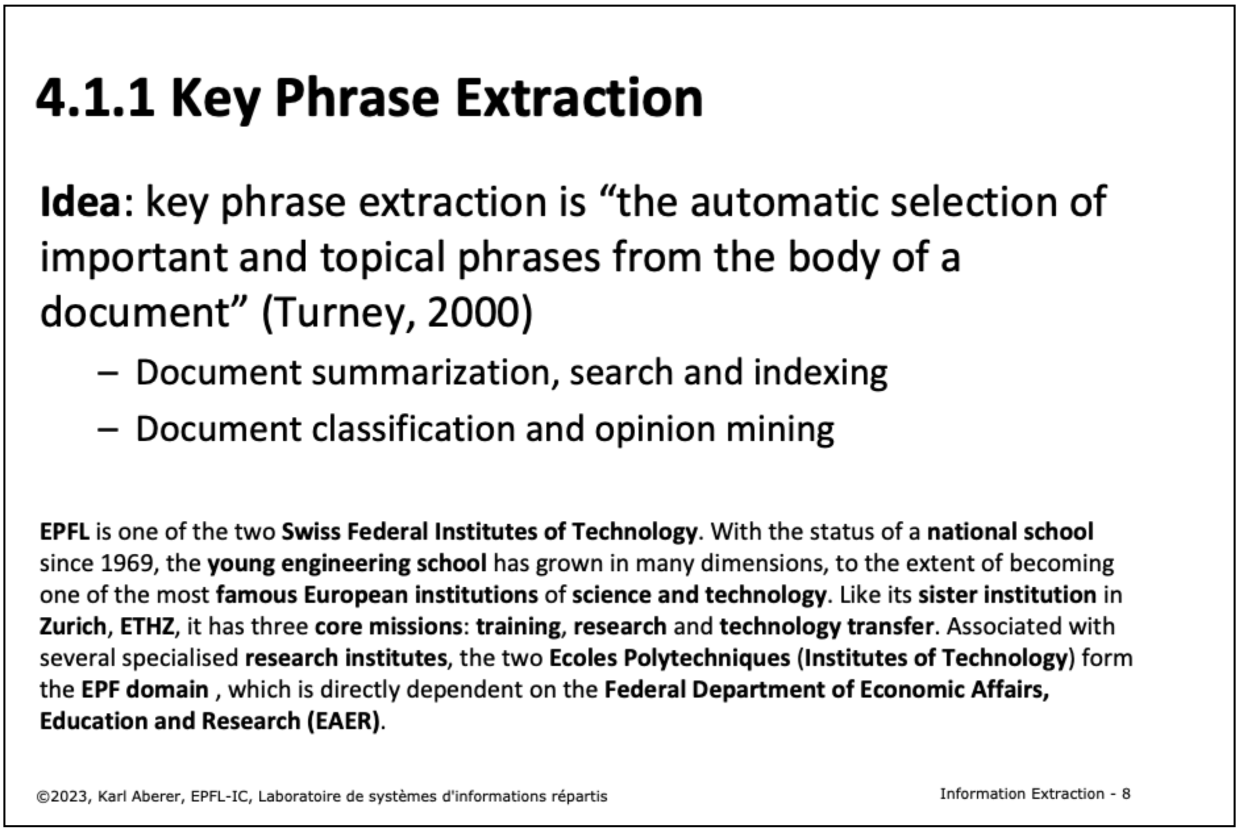
From text to knowledge

Who are the entities?

What are their attributes?

How are they related?

For investigating knowledge extraction, more commonly called information extraction, from textual content, we can consider the different constituents of a knowledge graph separately: entities, attributes and relationships. We will now introduce methods for extracting entities and then for establishing relationships among entities and with attributes.



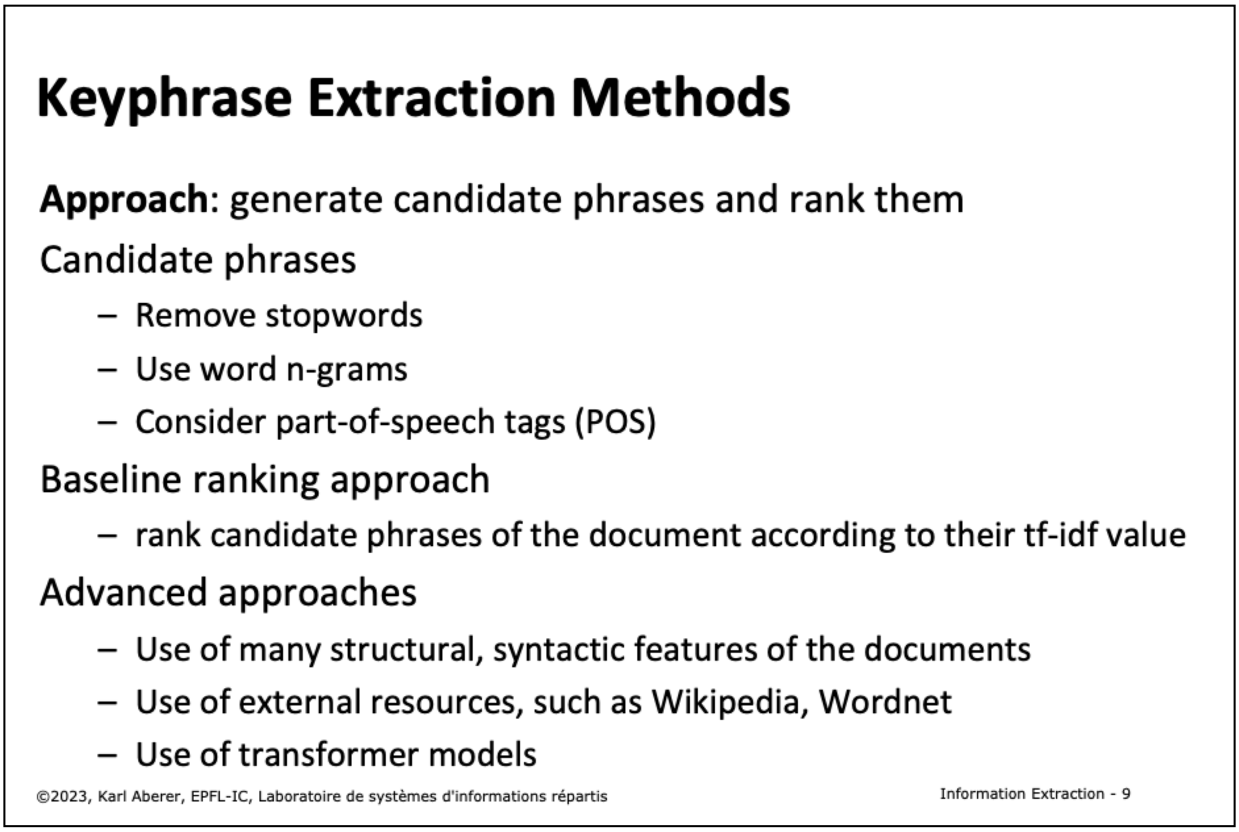
4.1.1 Key Phrase Extraction

Idea: key phrase extraction is "the automatic selection of important and topical phrases from the body of a document" (Turney, 2000)

- Document summarization, search and indexing
- Document classification and opinion mining

EPFL is one of the two **Swiss Federal Institutes of Technology**. With the status of a **national school** since 1969, the **young engineering school** has grown in many dimensions, to the extent of becoming one of the most **famous European institutions** of **science and technology**. Like its **sister institution** in **Zurich**, **ETHZ**, it has three **core missions**: **training**, **research** and **technology transfer**. Associated with several specialised **research institutes**, the two **Ecoles Polytechniques (Institutes of Technology**) form the **EPF domain** , which is directly dependent on the **Federal Department of Economic Affairs, Education and Research (EAER)**.

A first type of information extraction method is key phrase extraction. Key phrase extraction aims at identifying words and phrases that are typical for a document and characterize important concepts that occur in a document. Key phrase extraction has been developed to support document summarization, where the key phrase gives an overview of the key concepts of a document. Key phrase extraction supports also document search and indexing, where key phrases are used to index documents. Moreover, key phrases can also provide useful features for document classification, i.e., key phrase extraction can be considered as a feature selection method. In the example text we see the possible outcome of key phrase extraction, with all identified key phrases marked in bold.



Keyphrase Extraction Methods

Approach: generate candidate phrases and rank them

Candidate phrases

- Remove stopwords
- Use word n-grams
- Consider part-of-speech tags (POS)

Baseline ranking approach

- rank candidate phrases of the document according to their tf-idf value

Advanced approaches

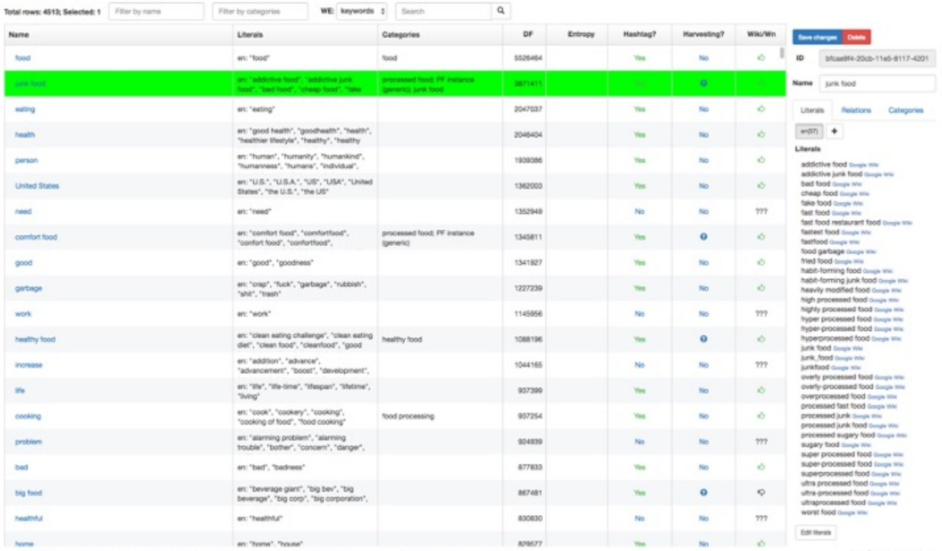
- Use of many structural, syntactic features of the documents
- Use of external resources, such as Wikipedia, Wordnet
- Use of transformer models

The basic approach to key phrase extraction uses principles well known from information retrieval. As in information retrieval stopwords are excluded. Key phrase candidates can be all word n-grams in the remaining text. Furthermore, part-of-speech tags can be used to further select the candidates, e.g., for excluding all verb phrases.

In order to assess whether a candidate phrase is characteristic for a document, tf-idf ranking is a possible approach. As in IR a candidate phrase is considered as relevant for a document if it is at the same time frequent and specific. Apart from that, many heuristics have been developed to refine this approach, considering additional features of the document. For example, a phrase in the title or a header could be considered as more relevant. External knowledge bases could be used to check whether a phrase corresponds to a commonly known concept. Recently, also transformer models have been applied for the task of key phrase extraction.

Use of Keyphrase Extraction

Creation of domain-specific thesaurus and taxonomy



Total rows: 4513; Selected: 1

Name	Literals	Categories	DF	Entropy	Hashtag?	Harvesting?	Wiki/Wn
food	en: "food"	food	5526464		Yes	No	👍
junk food	en: "addictive food", "addictive junk food", "bad food", "cheap food", "fake	processed food; PF instance (generic); junk food	3871411		Yes	❓	👍
eating	en: "eating"		2047037		Yes	No	👍
health	en: "good health", "goodhealth", "health", "healthier lifestyle", "healthy", "healthy		2046404		Yes	No	👍
person	en: "human", "humanity", "humankind", "humanness", "humans", "individual",		1939386		Yes	No	👍
United States	en: "U.S.", "U.S.A.", "US", "USA", "United States", "the U.S.", "the US"		1362003		Yes	No	👍
need	en: "need"		1352949		No	No	???
comfort food	en: "comfort food", "comfortfood", "confort food", "confortfood",	processed food; PF instance (generic)	1345811		Yes	❓	👍
good	en: "good", "goodness"		1341927		Yes	No	👍
garbage	en: "crap", "fuck", "garbage", "rubbish", "shit", "trash"		1227239		Yes	No	👍
work	en: "work"		1145956		No	No	???
healthy food	en: "clean eating challenge", "clean eating diet", "clean food", "cleanfood", "good	healthy food	1088196		Yes	❓	👍
increase	en: "addition", "advance", "advancement", "boost", "development",		1044165		No	No	???
life	en: "life", "life-time", "lifespan", "lifetime", "living"		937399		Yes	No	👍
cooking	en: "cook", "cookery", "cooking", "cooking of food", "food cooking"	food processing	937254		Yes	No	👍
problem	en: "alarming problem", "alarming trouble", "bother", "concern", "danger",		924939		No	No	???
bad	en: "bad", "badness"		877833		Yes	No	👍
big food	en: "beverage giant", "big bev", "big beverage", "big corp", "big corporation",		867481		Yes	❓	👎
healthful	en: "healthful"		830830		No	No	???
home	en: "home", "house"		829677		Yes	No	👍

ID: bfcae8f4-20cb-11e6-8117-4201

Name: junk food

Literals

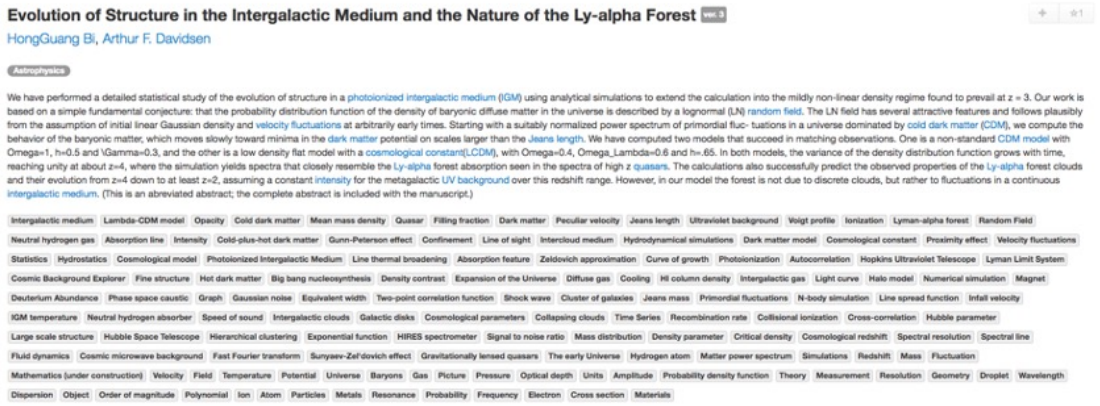
addictive food Google Wiki
addictive junk food Google Wiki
bad food Google Wiki
cheap food Google Wiki
fake food Google Wiki
fast food Google Wiki
fast food restaurant food Google Wiki
fastest food Google Wiki
fastfood Google Wiki
food garbage Google Wiki
fried food Google Wiki
habit-forming food Google Wiki
habit-forming junk food Google Wiki
heavily modified food Google Wiki
high processed food Google Wiki
highly processed food Google Wiki
hyper processed food Google Wiki
hyper-processed food Google Wiki
hyperprocessed food Google Wiki
junk food Google Wiki
junk_food Google Wiki
junkfood Google Wiki
overly processed food Google Wiki
overly-processed food Google Wiki
overprocessed food Google Wiki
processed fast food Google Wiki
processed junk Google Wiki
processed junk food Google Wiki
processed sugary food Google Wiki
sugary food Google Wiki
super processed food Google Wiki
super-processed food Google Wiki
superprocessed food Google Wiki
ultra processed food Google Wiki
ultra-processed food Google Wiki
ultraprocessed food Google Wiki
worst food Google Wiki

Key phrase extraction can be used as an initial step to create a domain-specific thesaurus or taxonomy. In the example, we see a thesaurus that has been constructed for the food domain, based on key phrase extraction. For the phrase “junk food” many synonyms and near-synonyms have been identified. In practice a human expert would not be able to extract reliably all different types of mentions of such a concept.

As a result, this allows to increase recall for retrieval of documents that refer to this concept.

Use of Keyphrase Extraction

Document classification and search



Evolution of Structure in the Intergalactic Medium and the Nature of the Ly-alpha Forest ver. 3

HongGuang Bi, Arthur F. Davidsen

Astrophysics

We have performed a detailed statistical study of the evolution of structure in a photoionized intergalactic medium (IGM) using analytical simulations to extend the calculation into the mildly non-linear density regime found to prevail at z = 3. Our work is based on a simple fundamental conjecture: that the probability distribution function of the density of baryonic diffuse matter in the universe is described by a lognormal (LN) random field. The LN field has several attractive features and follows plausibly from the assumption of initial linear Gaussian density and velocity fluctuations at arbitrarily early times. Starting with a suitably normalized power spectrum of primordial fluc- tuations in a universe dominated by cold dark matter (CDM), we compute the behavior of the baryonic matter, which moves slowly toward minima in the dark matter potential on scales larger than the Jeans length. We have computed two models that succeed in matching observations. One is a non-standard CDM model with Omega=1, h=0.5 and \Gamma=0.3, and the other is a low density flat model with a cosmological constant(LCDM), with Omega=0.4, Omega_Lambda=0.6 and h=.65. In both models, the variance of the density distribution function grows with time, reaching unity at about z=4, where the simulation yields spectra that closely resemble the Ly-alpha forest absorption seen in the spectra of high z quasars. The calculations also successfully predict the observed properties of the Ly-alpha forest clouds and their evolution from z=4 down to at least z=2, assuming a constant intensity for the metagalactic UV background over this redshift range. However, in our model the forest is not due to discrete clouds, but rather to fluctuations in a continuous intergalactic medium. (This is an abreviated abstract; the complete abstract is included with the manuscript.)

Intergalactic medium | Lambda-CDM model | Opacity | Cold dark matter | Mean mass density | Quasar | Filling fraction | Dark matter | Peculiar velocity | Jeans length | Ultraviolet background | Voigt profile | Ionization | Lyman-alpha forest | Random Field

Neutral hydrogen gas | Absorption line | Intensity | Cold-plus-hot dark matter | Gunn-Peterson effect | Confinement | Line of sight | Intercloud medium | Hydrodynamical simulations | Dark matter model | Cosmological constant | Proximity effect | Velocity fluctuations

Statistics | Hydrostatics | Cosmological model | Photoionized Intergalactic Medium | Line thermal broadening | Absorption feature | Zeldovich approximation | Curve of growth | Photoionization | Autocorrelation | Hopkins Ultraviolet Telescope | Lyman Limit System

Cosmic Background Explorer | Fine structure | Hot dark matter | Big bang nucleosynthesis | Density contrast | Expansion of the Universe | Diffuse gas | Cooling | HI column density | Intergalactic gas | Light curve | Halo model | Numerical simulation | Magnet

Deuterium Abundance | Phase space caustic | Graph | Gaussian noise | Equivalent width | Two-point correlation function | Shock wave | Cluster of galaxies | Jeans mass | Primordial fluctuations | N-body simulation | Line spread function | Infall velocity

IGM temperature | Neutral hydrogen absorber | Speed of sound | Intergalactic clouds | Galactic disks | Cosmological parameters | Collapsing clouds | Time Series | Recombination rate | Collisional ionization | Cross-correlation | Hubble parameter

Large scale structure | Hubble Space Telescope | Hierarchical clustering | Exponential function | HIRES spectrometer | Signal to noise ratio | Mass distribution | Density parameter | Critical density | Cosmological redshift | Spectral resolution | Spectral line

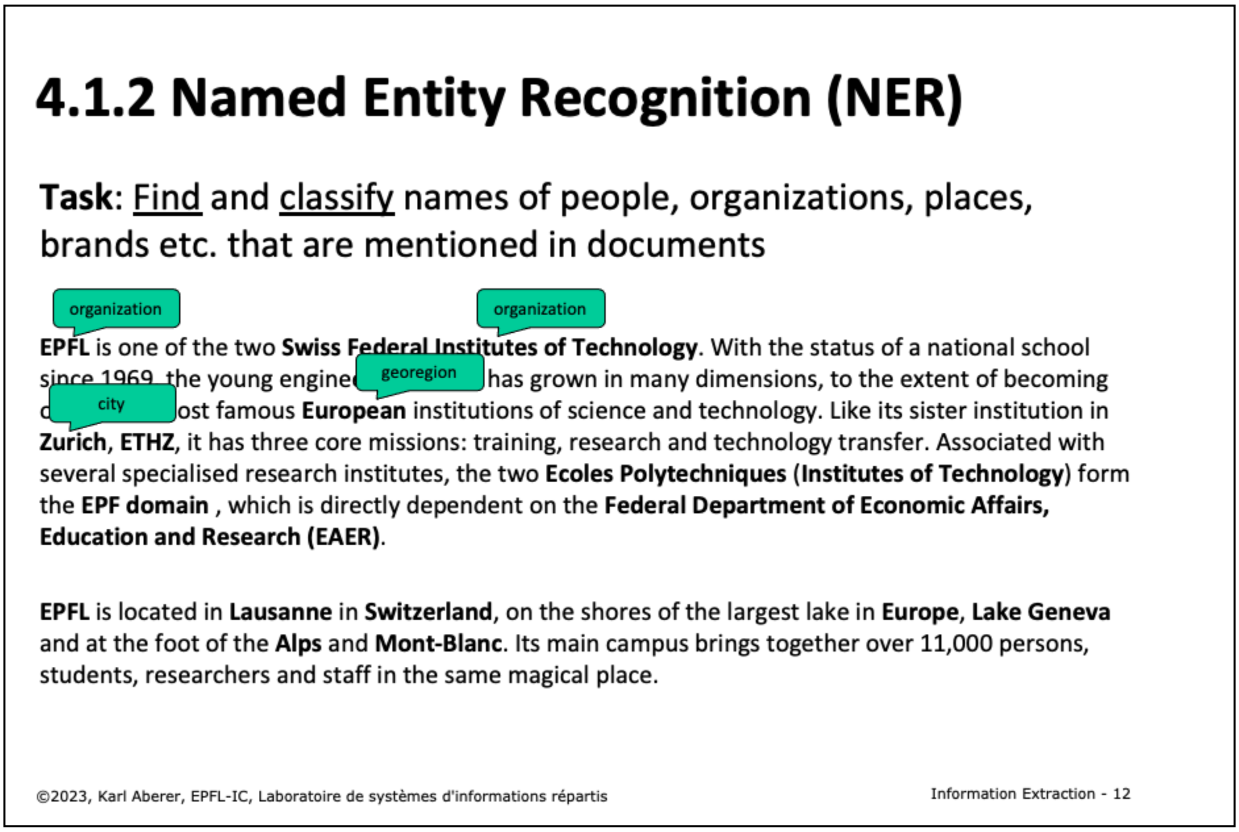
Fluid dynamics | Cosmic microwave background | Fast Fourier transform | Sunyaev-Zel'dovich effect | Gravitationally lensed quasars | The early Universe | Hydrogen atom | Matter power spectrum | Simulations | Redshift | Mass | Fluctuation

Mathematics (under construction) | Velocity | Field | Temperature | Potential | Universe | Baryons | Gas | Picture | Pressure | Optical depth | Units | Amplitude | Probability density function | Theory | Measurement | Resolution | Geometry | Droplet | Wavelength

Dispersion | Object | Order of magnitude | Polynomial | Ion | Atom | Particles | Metals | Resonance | Probability | Frequency | Electron | Cross section | Materials

sciencewise.info

This is an example of using key phrase extraction for scientific documents in physics. Many highly specific concepts are identified automatically and allow a scientist to more precisely filter and search the documents. You can try this out on your own at sciencewise.info.



4.1.2 Named Entity Recognition (NER)

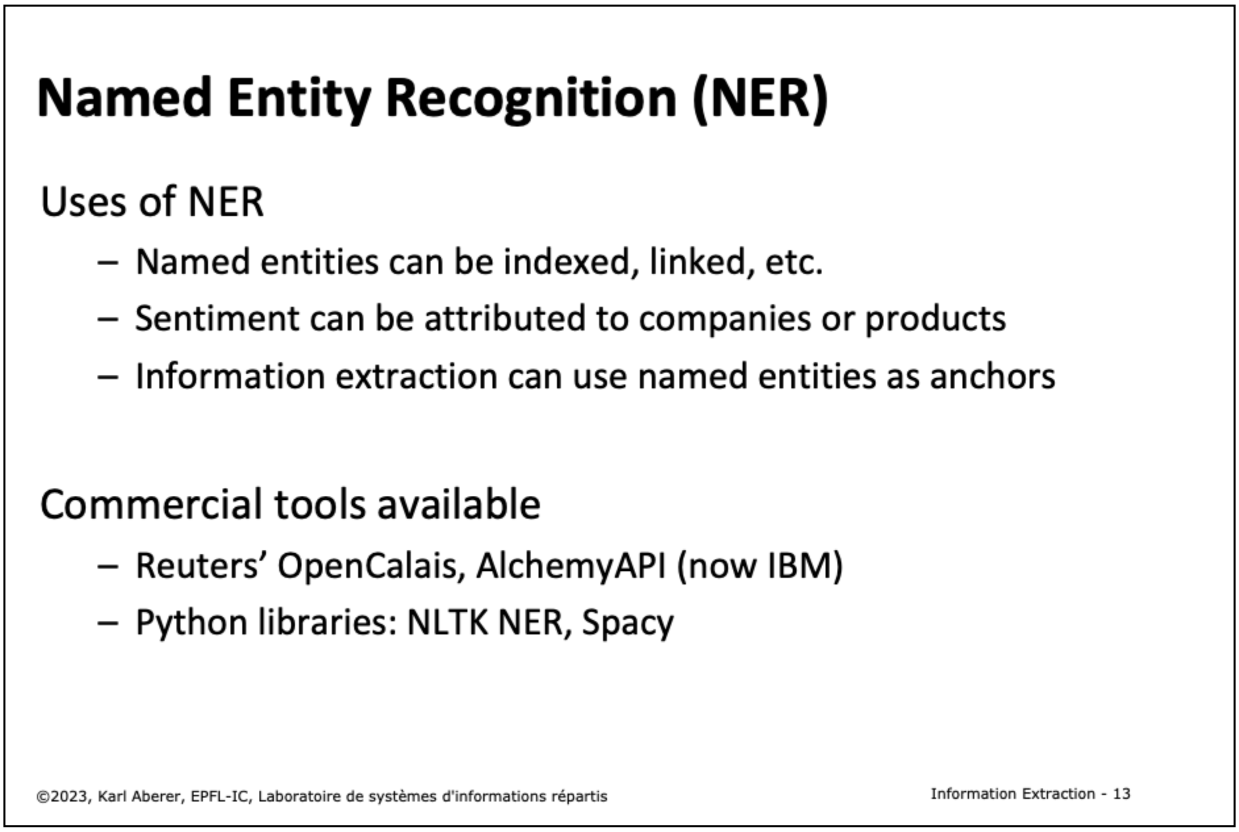
Task: Find and classify names of people, organizations, places, brands etc. that are mentioned in documents

organization
organization
georegion
city

EPFL is one of the two **Swiss Federal Institutes of Technology**. With the status of a national school since 1969, the young engine has grown in many dimensions, to the extent of becoming o ost famous **European** institutions of science and technology. Like its sister institution in **Zurich**, **ETHZ**, it has three core missions: training, research and technology transfer. Associated with several specialised research institutes, the two **Ecoles Polytechniques** (**Institutes of Technology**) form the **EPF domain** , which is directly dependent on the **Federal Department of Economic Affairs, Education and Research (EAER)**.

EPFL is located in **Lausanne** in **Switzerland**, on the shores of the largest lake in **Europe**, **Lake Geneva** and at the foot of the **Alps** and **Mont-Blanc**. Its main campus brings together over 11,000 persons, students, researchers and staff in the same magical place.

Named entity recognition is a more specific task than key phrase extraction. In NER the objective is to identify phrases that are names of specific types of entities, such as people, organizations or places. This, again, is very useful for document classification and search, but also a steppingstone to extract more complex knowledge, in particular statements relating different entities, as we will see later.



Named Entity Recognition (NER)

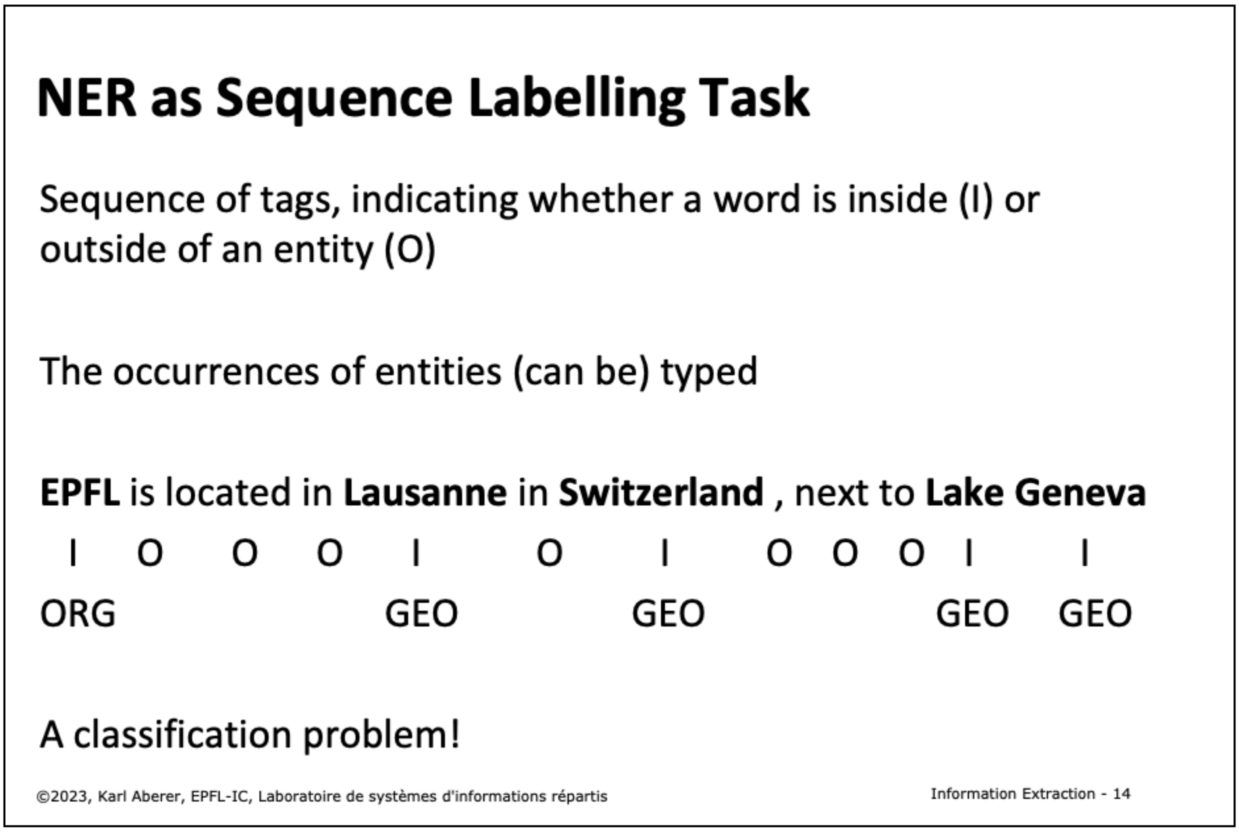
Uses of NER

- Named entities can be indexed, linked, etc.
- Sentiment can be attributed to companies or products
- Information extraction can use named entities as anchors

Commercial tools available

- Reuters’ OpenCalais, AlchemyAPI (now IBM)
- Python libraries: NLTK NER, Spacy

NER has many commercial applications, e.g., for marketing or studying public perception, by linking volume of communication, sentiment and popularity to specific entities, such as products, companies or organizations. Thus, there exist manycommercial tools that offer this type of service.



NER as Sequence Labelling Task

Sequence of tags, indicating whether a word is inside (I) or outside of an entity (O)

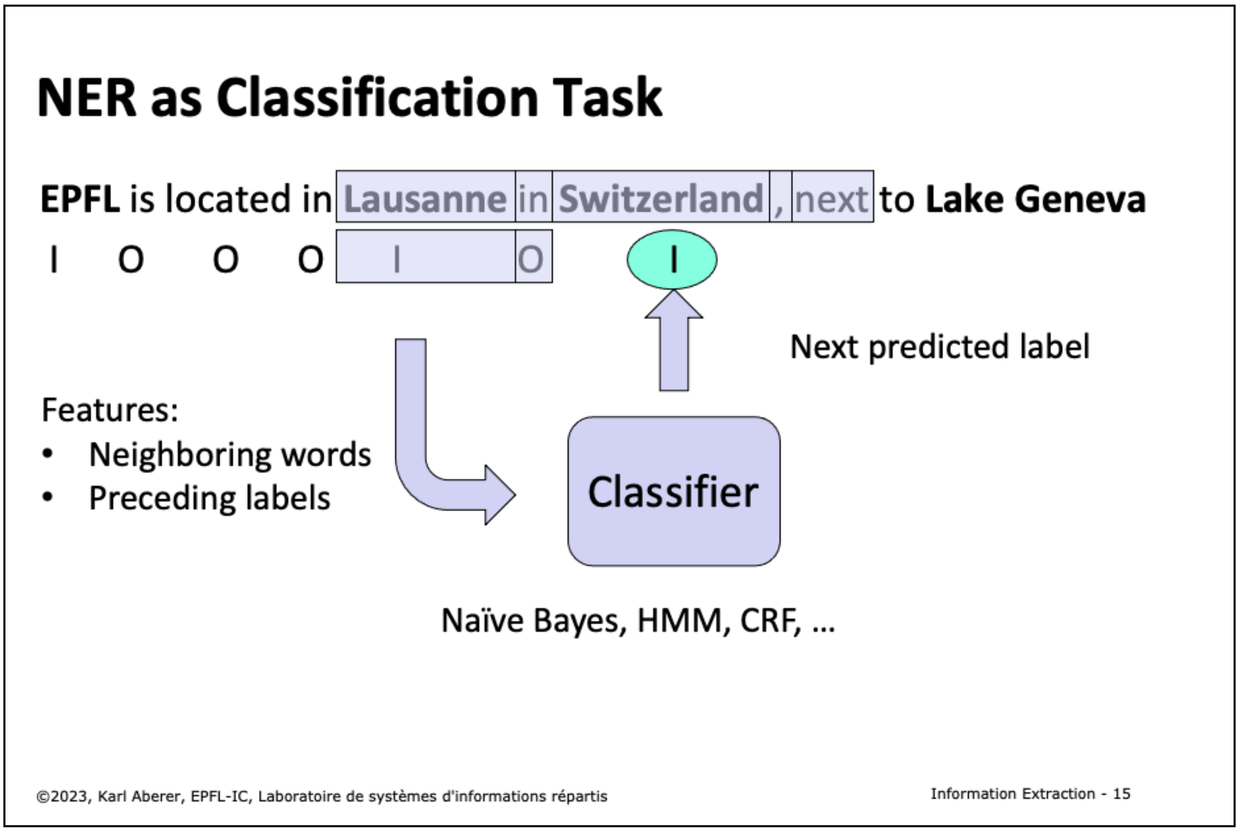
The occurrences of entities (can be) typed

EPFL	is	located	in	**Lausanne**	in	**Switzerland**	,	next	to	**Lake**	**Geneva**
I	O	O	O	I	O	I	O	O	O	I	I
ORG				GEO		GEO				GEO	GEO

A classification problem!

The basic task of NER is to detect whether a word belongs to an entity name or not. Furthermore, when an entity name is detected, it can be classified according to the type of the entity, e.g., an organisation (ORG), a location (GEO), a person etc.

When analyzing a text, NER is thus a classification problem, where for each word it needs to be decided whether it is inside or outside of an entity name. More detailed classifications, whether a word is the beginning or end of entity name, can also be performed. Note that in this context also punctuation marks are considered as words, as they may carry important information on the presence of an entity.



NER as Classification Task

EPFL is located in **Lausanne** in **Switzerland** , next to **Lake Geneva**

I O O O I O I

Next predicted label

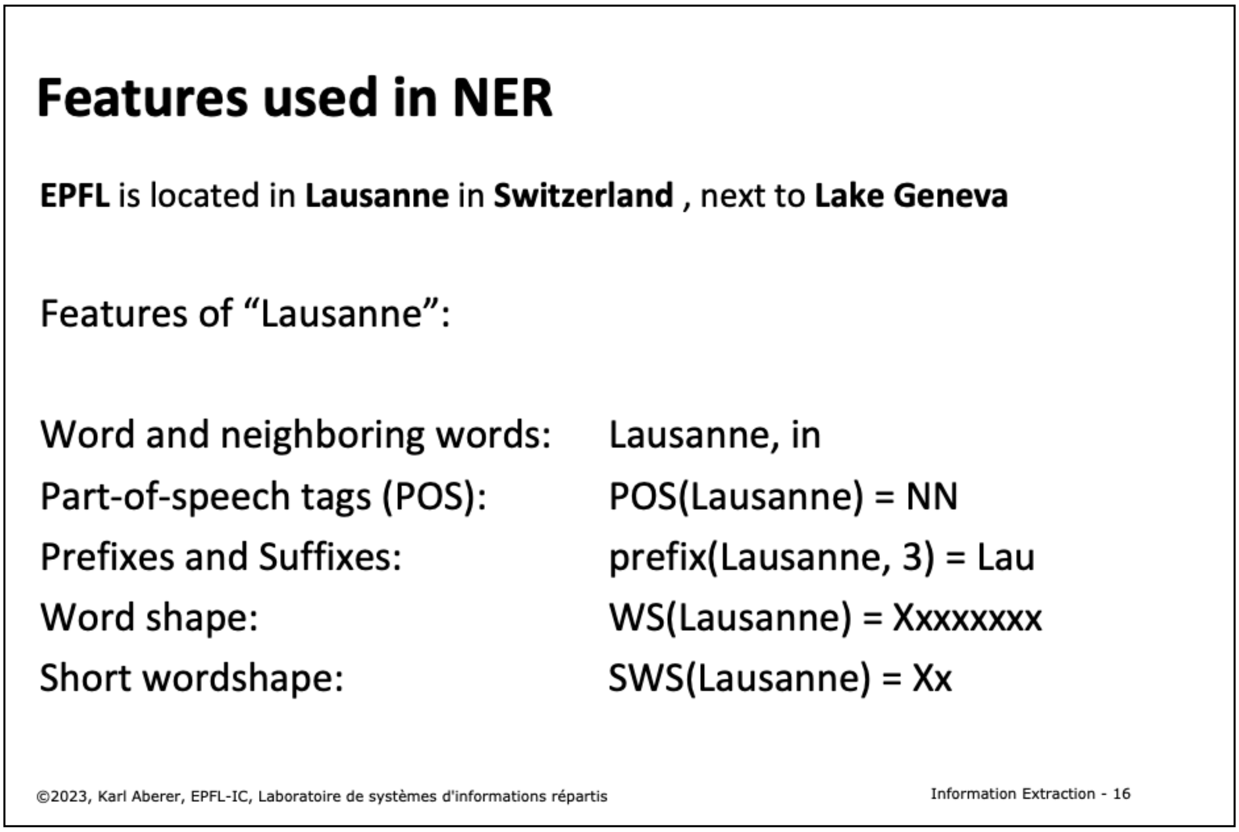
Features:

- Neighboring words
- Preceding labels

Classifier

Naïve Bayes, HMM, CRF, …

Given that NER can be considered as a classification problem, we must decide on two questions. First, which are the input features for the classifier, and second, which is the classification algorithm to be used. As for the input features, typically the neighborhood of a word is considered. In this neighborhood we find other words, which can be used as directly as features and from which several derived features can be produced. The classifier classifies words while reading the words in the sequence they appear in the document. Therefore, one special kind of feature that is used in NER are the labels that have been produced by the classifier for words preceding the word to be classified. Even though in principle any classification algorithm could be applied, e.g., Naïve Bayes, specific sequence-oriented classifiers (HMM, CRF) can have better performance.



Features used in NER

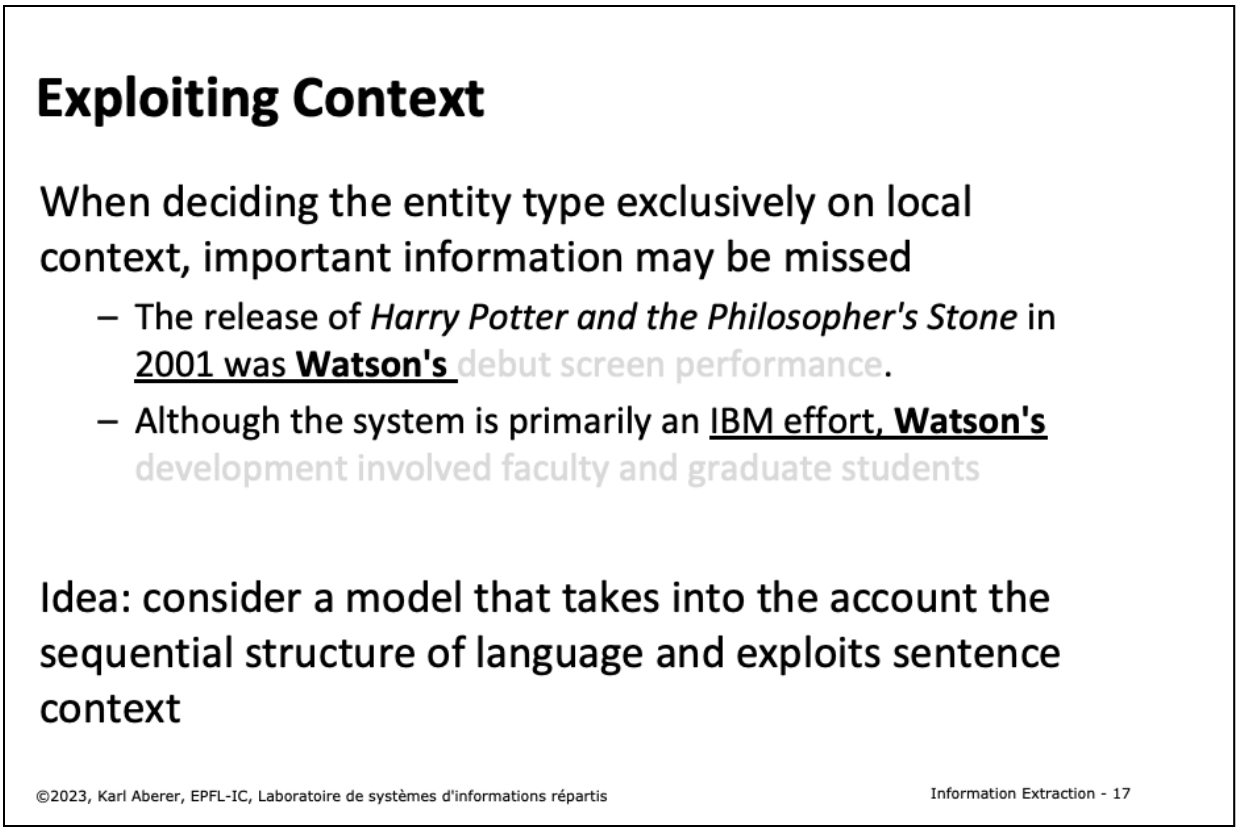
EPFL is located in **Lausanne** in **Switzerland** , next to **Lake Geneva**

Features of “Lausanne”:

Word and neighboring words:	Lausanne, in
Part-of-speech tags (POS):	POS(Lausanne) = NN
Prefixes and Suffixes:	prefix(Lausanne, 3) = Lau
Word shape:	WS(Lausanne) = Xxxxxxxx
Short wordshape:	SWS(Lausanne) = Xx

Information Extraction - 16

Here we see a list of typical features that are used in named entity recognition. Some of them are quite specific to the task. For example, part-of-speech tags can be helpful as they allow to distinguish noun phrases (NN) which are typical for entities. Pre- and suffixes are another interesting feature. For example, words ending in “land” would often be locations. Learning this fact can help to generalize the classification to new terms that would contain such a suffix. For entities, in particular in English, also the word shape is an important feature, as usually proper names start with capital letters, or acronyms consist of capital letters only.



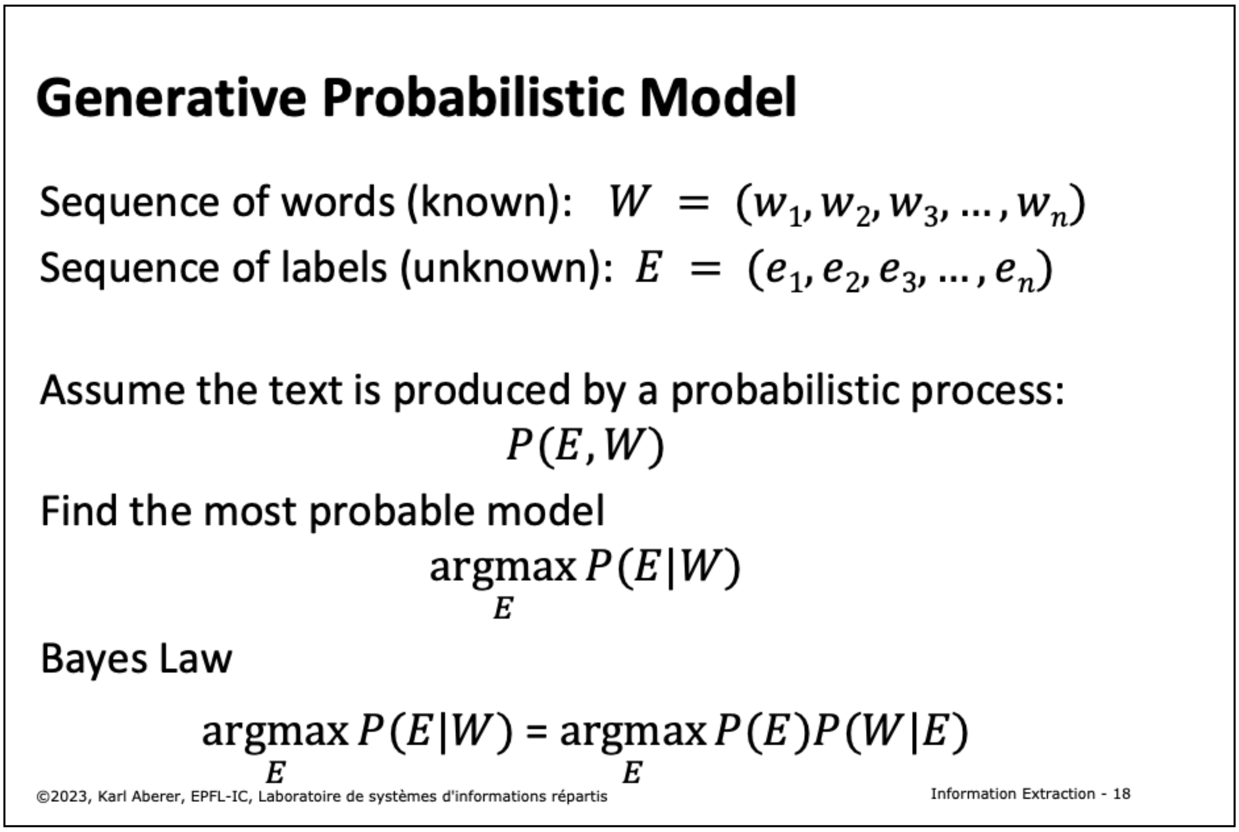
Exploiting Context

When deciding the entity type exclusively on local context, important information may be missed

- The release of *Harry Potter and the Philosopher's Stone* in 2001 was **Watson's** debut screen performance.
- Although the system is primarily an IBM effort, **Watson's** development involved faculty and graduate students

Idea: consider a model that takes into the account the sequential structure of language and exploits sentence context

If only features derived from the local context from the immediate neighborhood of a word to be classified are used, important context information can be missed. This motivated the use of classification models that exploit the larger context of a word in the text, like the complete sentence in which the word occurs.



Generative Probabilistic Model

Sequence of words (known): $W = (w_1, w_2, w_3, \dots, w_n)$
Sequence of labels (unknown): $E = (e_1, e_2, e_3, \dots, e_n)$

Assume the text is produced by a probabilistic process:

$$P(E, W)$$

Find the most probable model

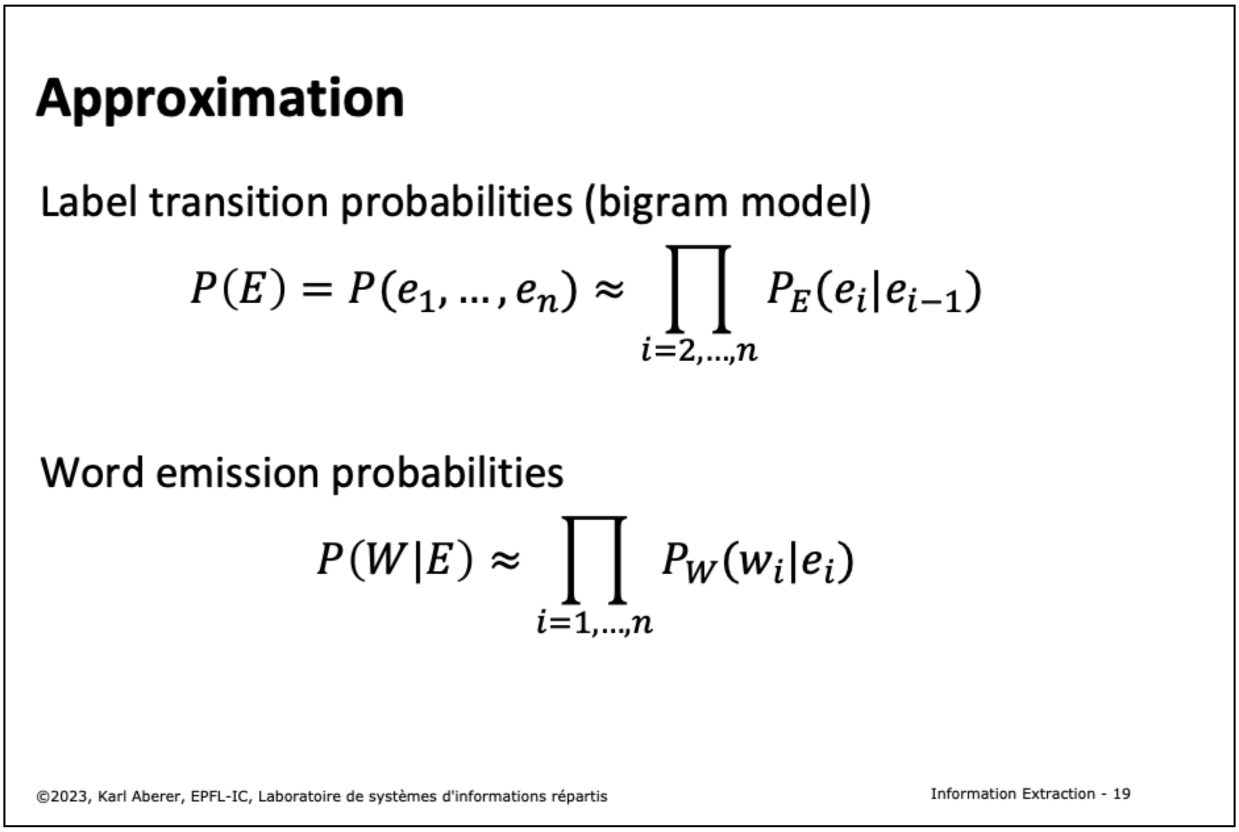
$$\underset{E}{\operatorname{argmax}} P(E|W)$$

Bayes Law

$$\underset{E}{\operatorname{argmax}} P(E|W) = \underset{E}{\operatorname{argmax}} P(E)P(W|E)$$

We introduce now a classification approach that has been used for NER and exploits the sequential nature of natural language. The approach belongs to the class of generative probabilistic models, like the one we have introduced for information retrieval.

The basic model assumes that there exists a (unknown) probability distribution $P(E, W)$ that correlates sequences of words with the corresponding sequences of entity labels. The classification task is then to identify for a given sequence of words, the most probable sequence of labels. Using Bayes law, we can reformulate this, by decomposing the conditional probability $P(E|W)$ into the product of two probability distributions. $P(E)$ is a model describing of the probability of different labels to occur, and $P(W|E)$ is a model describing of how words correlate with labels.



Approximation

Label transition probabilities (bigram model)

$$P(E) = P(e_1, \dots, e_n) \approx \prod_{i=2,\dots,n} P_E(e_i | e_{i-1})$$

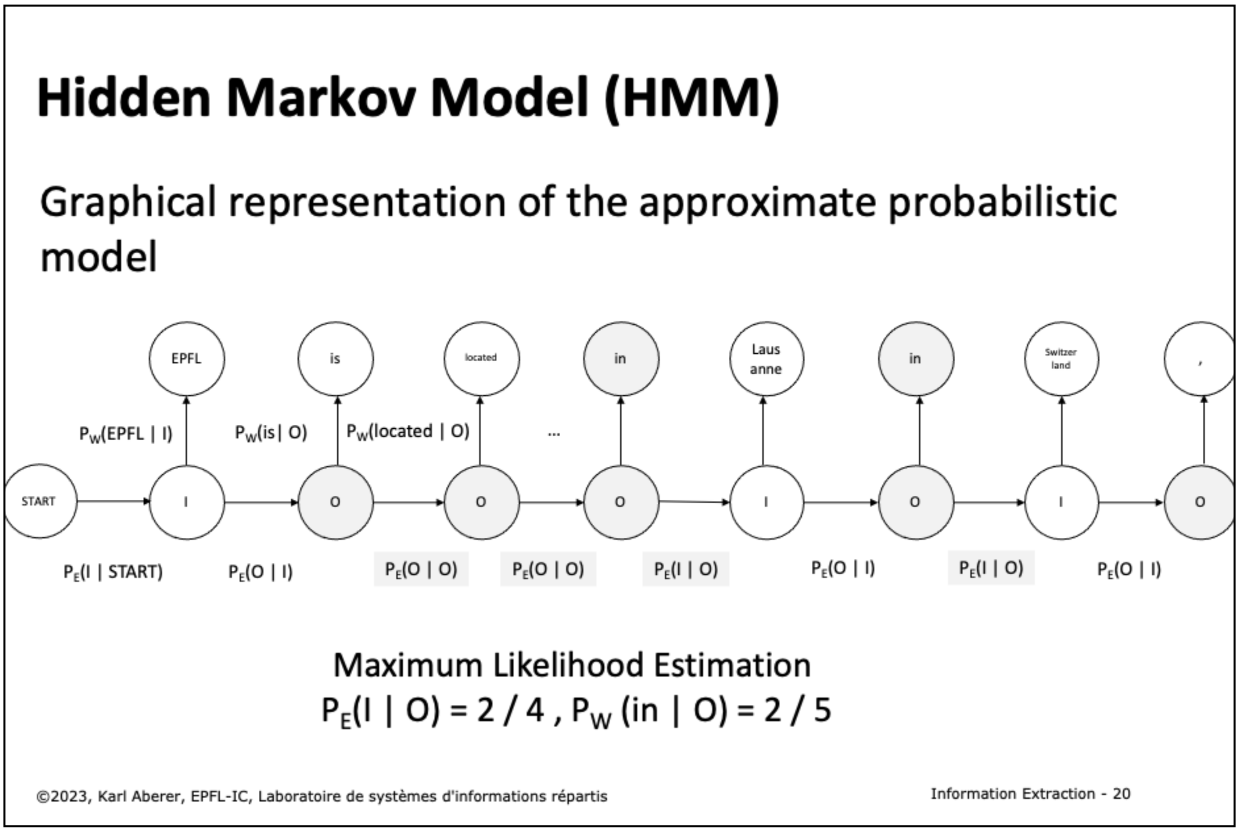
Word emission probabilities

$$P(W|E) \approx \prod_{i=1,\dots,n} P_W(w_i | e_i)$$

©2023, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Information Extraction - 19

As it is not possible to estimate the complete probability distribution functions $P(E)$ and $P(E|W)$, we approximate them by making independence assumptions. We assume that the probability of a label to occur, depends only on the previous label. This corresponds to a bigram model, generalizing the unigram model we have introduced in probabilistic information retrieval. For a word we assume that its probability of occurrence depends only on the label it received. Thus, the two probability functions decompose into products of simpler functions that we can estimate.



Hidden Markov Model (HMM)

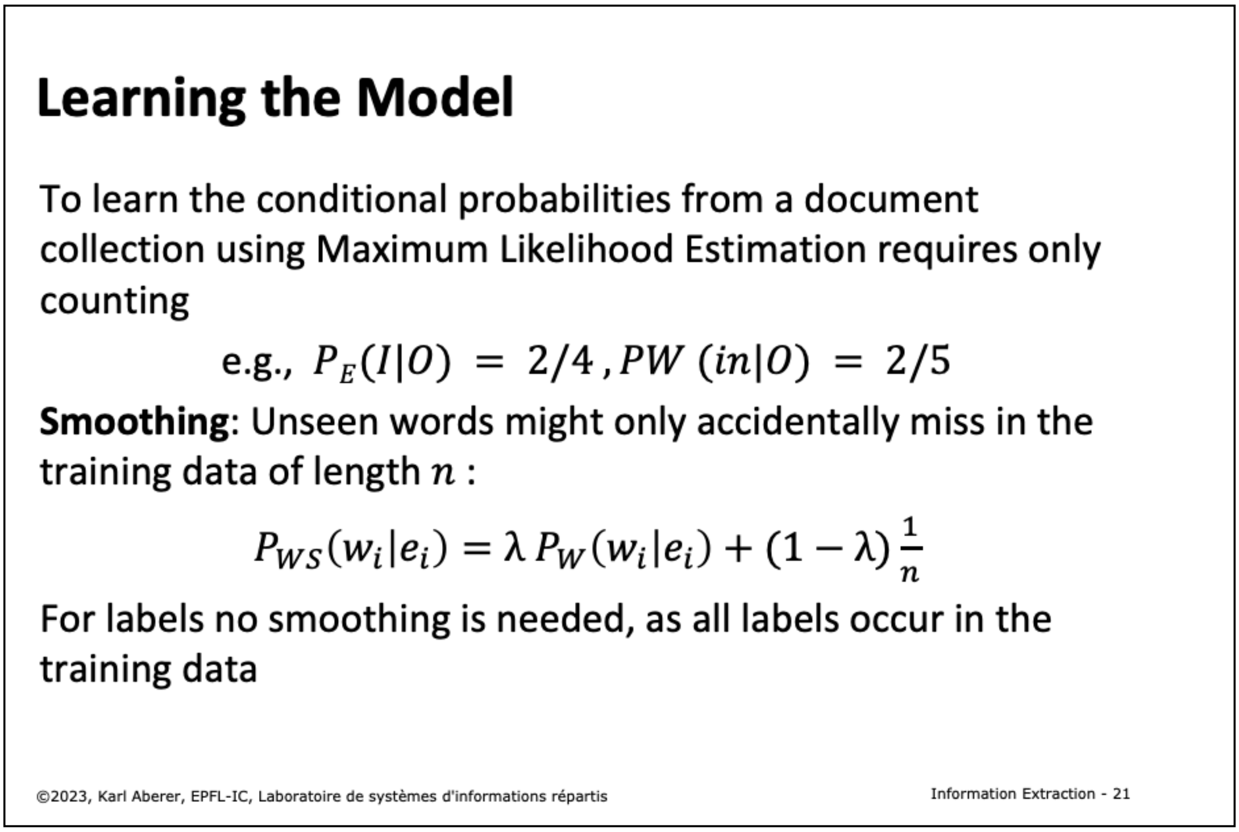
Graphical representation of the approximate probabilistic model

Maximum Likelihood Estimation

$P_E(I \mid O) = 2 / 4$, $P_W(\text{in} \mid O) = 2 / 5$

We can represent approximate model of the probability distribution graphically as a Markov Model, where we indicate which probabilistic variables depend on which others. More precisely, it is a Hidden Markov Model (HMM). The hidden labels E are unknown, and their probabilities need to be estimated from the words that can be observed.

This approach for estimating probabilities of hidden variables with HMMs can be applied to other sequence labelling tasks. For example, it can be used to learn part-of-speech tags and the types of the entities.



Learning the Model

To learn the conditional probabilities from a document collection using Maximum Likelihood Estimation requires only counting

e.g., $P_E(I|O) = 2/4$, $PW\ (in|O) = 2/5$

Smoothing: Unseen words might only accidentally miss in the training data of length n :

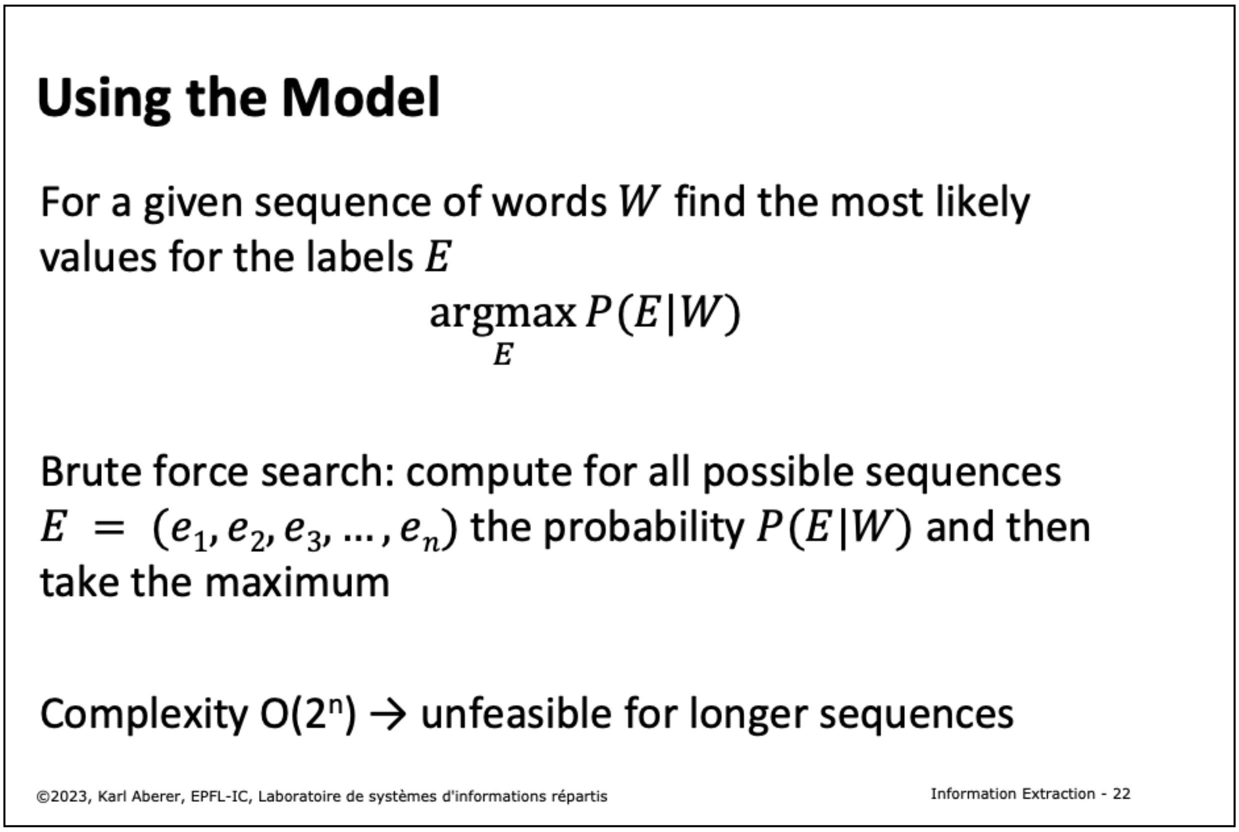
$$P_{WS}(w_i|e_i) = \lambda\, P_W(w_i|e_i) + (1 - \lambda)\frac{1}{n}$$

For labels no smoothing is needed, as all labels occur in the training data

Given a document collection we can estimate the probabilities $P_E(e_i|e_{i-1})$ and $P_W(w_i|e_i)$. As in probabilistic information retrieval we use maximum likelihood estimation. For example, for estimating the probability $P_E(e_i|O)$ we count the total number of occurrences of O, and then compute the ratio between the cases where the preceding label is e_i with the total number of occurrences of O.

As in probabilistic information retrieval, we need to consider the issue of sparsity of words in the training set. It might be the case that a specific word does not occur together with a given label in the training data, whereas this word still might be related to the label in general. Therefore, smoothing is applied, where the smoothing parameter depends on the size of the training data. The more data is available, the less the likelihood that a word-label pair that is likely to occur is not found in the training data.

For estimating the probability of labels to occur, no smoothing is required, as the number of labels is very small and thus all pairs of labels combinations are likely to occur in the training data.



Using the Model

For a given sequence of words W find the most likely values for the labels E

$$\underset{E}{\operatorname{argmax}} P(E|W)$$

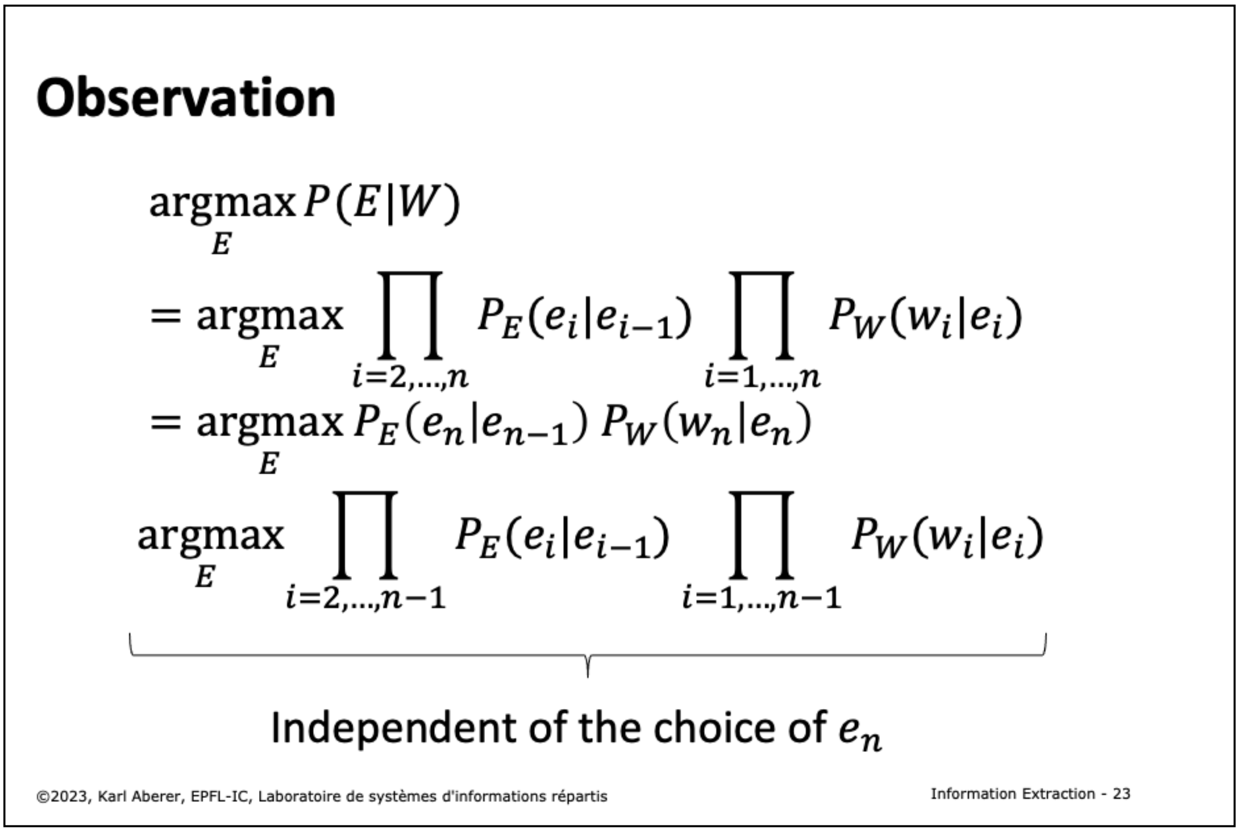
Brute force search: compute for all possible sequences $E = (e_1, e_2, e_3, \ldots, e_n)$ the probability $P(E|W)$ and then take the maximum

Complexity $O(2^n)$ → unfeasible for longer sequences

©2023, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Information Extraction - 22

Once the parameters of the HMM model have been derived from the training data, they can be used to estimate the most likely values of labels for an unknown sequence of words. One possibility is to apply brute-force search by computing the probability of each possible label sequence using the model. For longer sequences this becomes computationally intractable as the number of possible sequences grows exponentially.



Observation

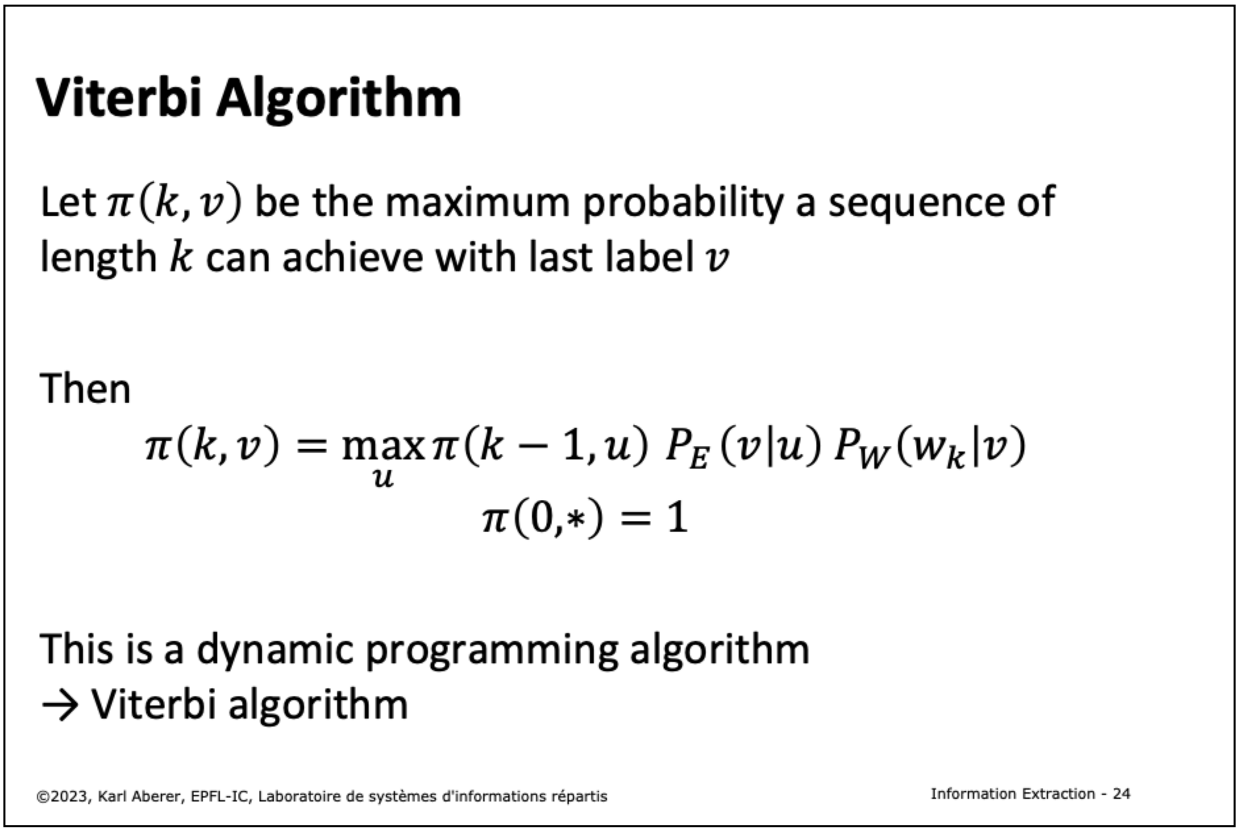
$$\underset{E}{\operatorname{argmax}} P(E|W)$$

$$= \underset{E}{\operatorname{argmax}} \prod_{i=2,\dots,n} P_E(e_i|e_{i-1}) \prod_{i=1,\dots,n} P_W(w_i|e_i)$$

$$= \underset{E}{\operatorname{argmax}} P_E(e_n|e_{n-1})\, P_W(w_n|e_n)$$

$$\underbrace{\underset{E}{\operatorname{argmax}} \prod_{i=2,\dots,n-1} P_E(e_i|e_{i-1}) \prod_{i=1,\dots,n-1} P_W(w_i|e_i)}_{\text{Independent of the choice of } e_n}$$

We made an independence assumption on the probabilities of the elements of a label sequence, where a label depends only on its predecessor in the sequence. We can exploit this independence assumption to simplify the computation of the sequence probability. The choice of the the last label in the sequence that maximizes the overall probability is independent of the choices of the other labels in the sequence maximizing the overall probability. Using this property simplifies the computation of the sequence probability significantly. Note tha



Viterbi Algorithm

Let $\pi(k, v)$ be the maximum probability a sequence of length k can achieve with last label v

Then

$$\pi(k, v) = \max_{u} \pi(k-1, u)\ P_E(v|u)\ P_W(w_k|v)$$
$$\pi(0,*) = 1$$

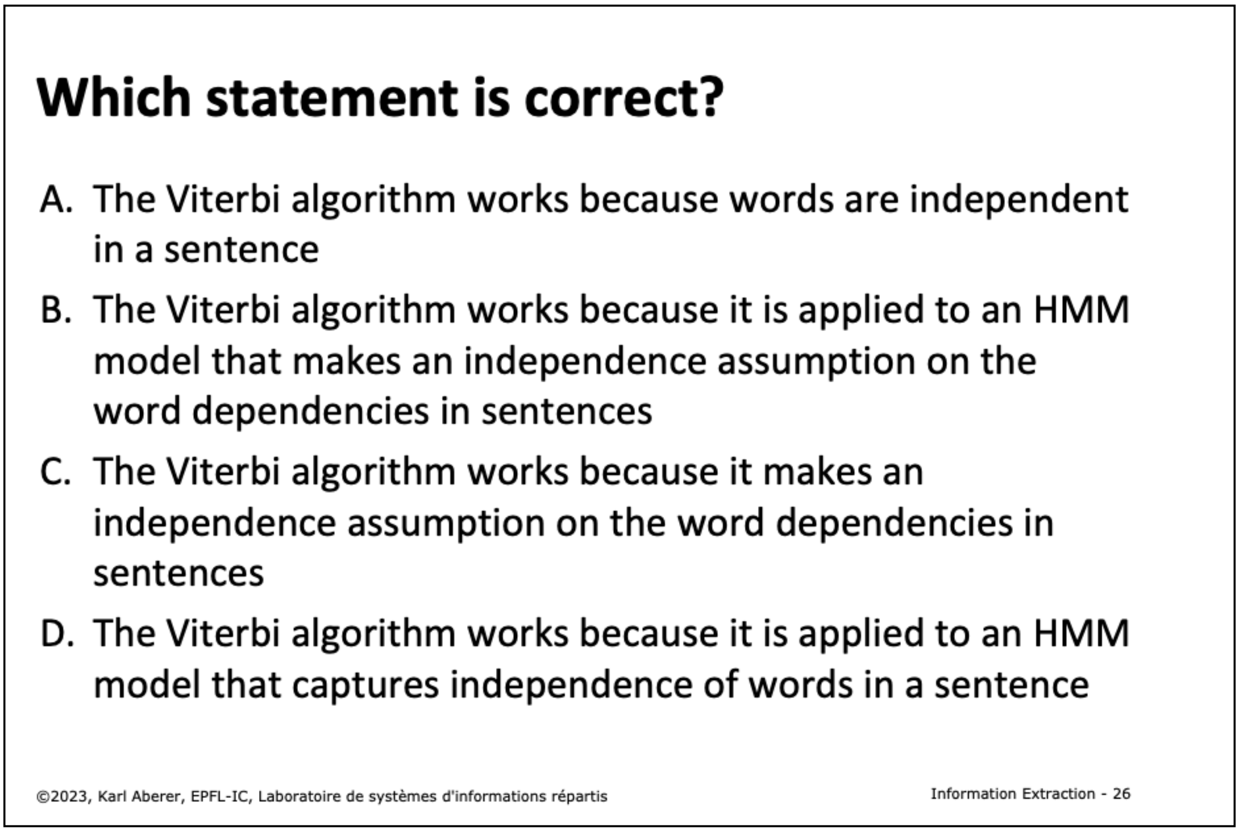
This is a dynamic programming algorithm
→ Viterbi algorithm

Using the independence assumption described before, we can iteratively compute the sequence of labels that maximizes the probability, using in each step the label sequence found so far. The maximum probability $\pi(k, v)$ that a label sequence of length k can achieve, if the last label is v can be computed from the maximum probabilities known for shorter sequences and the parameters of the probabilistic model.

This algorithm is a simple version of Viterbi's algorithm. In its general form a random variable can depend on several earlier random variables in the earlier sequence, resulting in a dynamic programming algorithm.

An HMM model would not be an appropriate approach to identify

A. Named Entities
B. Part-of-Speech tags
C. Concepts
D. Word n-grams



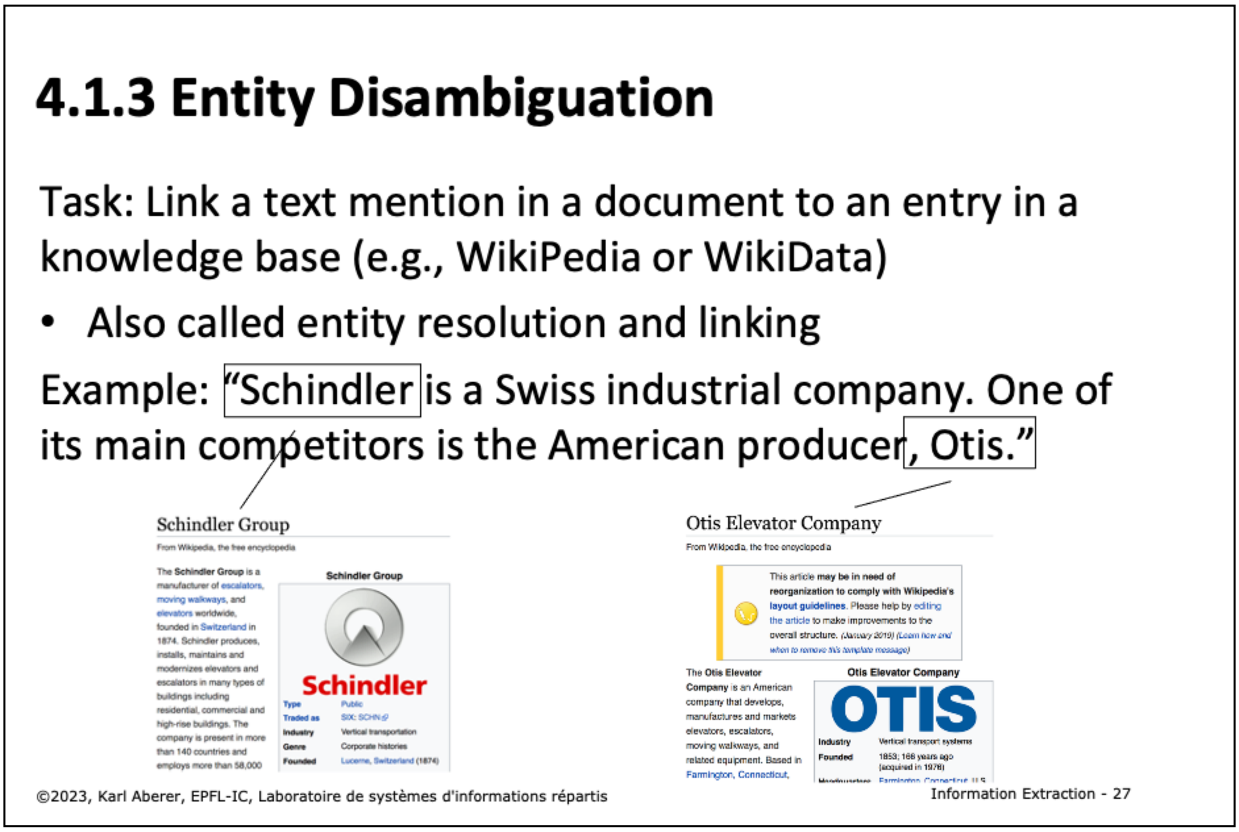
Which statement is correct?

A. The Viterbi algorithm works because words are independent in a sentence

B. The Viterbi algorithm works because it is applied to an HMM model that makes an independence assumption on the word dependencies in sentences

C. The Viterbi algorithm works because it makes an independence assumption on the word dependencies in sentences

D. The Viterbi algorithm works because it is applied to an HMM model that captures independence of words in a sentence



4.1.3 Entity Disambiguation

Task: Link a text mention in a document to an entry in a knowledge base (e.g., WikiPedia or WikiData)

- Also called entity resolution and linking

Example: "Schindler is a Swiss industrial company. One of its main competitors is the American producer, Otis."

Schindler Group

From Wikipedia, the free encyclopedia

The **Schindler Group** is a manufacturer of escalators, moving walkways, and elevators worldwide, founded in Switzerland in 1874. Schindler produces, installs, maintains and modernizes elevators and escalators in many types of buildings including residential, commercial and high-rise buildings. The company is present in more than 140 countries and employs more than 58,000

Schindler Group

Type	Public
Traded as	SIX: SCHN
Industry	Vertical transportation
Genre	Corporate histories
Founded	Lucerne, Switzerland (1874)

Otis Elevator Company

From Wikipedia, the free encyclopedia

This article **may be in need of reorganization to comply with Wikipedia's layout guidelines.** Please help by editing the article to make improvements to the overall structure. *(January 2019) (Learn how and when to remove this template message)*

The **Otis Elevator Company** is an American company that develops, manufactures and markets elevators, escalators, moving walkways, and related equipment. Based in Farmington, Connecticut,

Otis Elevator Company

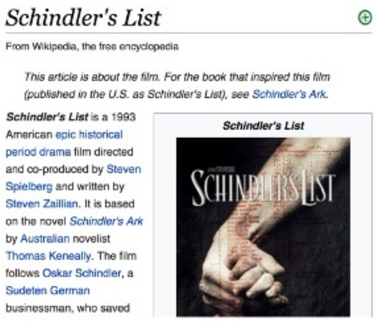
Industry	Vertical transport systems
Founded	1853; 168 years ago (acquired in 1976)
Headquarters	Farmington, Connecticut, U.S.

Once entities have been recognized in a text, one can link them to their corresponding counter-parts in a knowledge base. So-called entity disambiguation is a step that usually follows named entity recognition.

Challenge

Two problems

- Homonyms: entities with the same name
- Synonyms: different names for the same entity



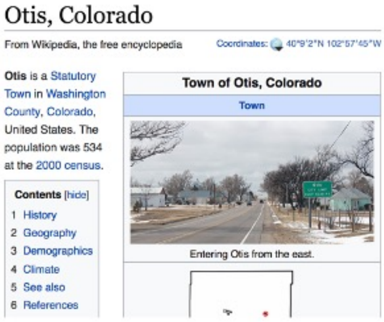
Schindler's List

From Wikipedia, the free encyclopedia

This article is about the film. For the book that inspired this film (published in the U.S. as Schindler's List), see Schindler's Ark.

Schindler's List is a 1993 American epic historical period drama film directed and co-produced by Steven Spielberg and written by Steven Zaillian. It is based on the novel *Schindler's Ark* by Australian novelist Thomas Keneally. The film follows Oskar Schindler, a Sudeten German businessman, who saved

Schindler's List



Otis, Colorado

From Wikipedia, the free encyclopedia

Coordinates: 40°9′2″N 102°57′45″W

Otis is a Statutory Town in Washington County, Colorado, United States. The population was 534 at the 2000 census.

Contents [hide]

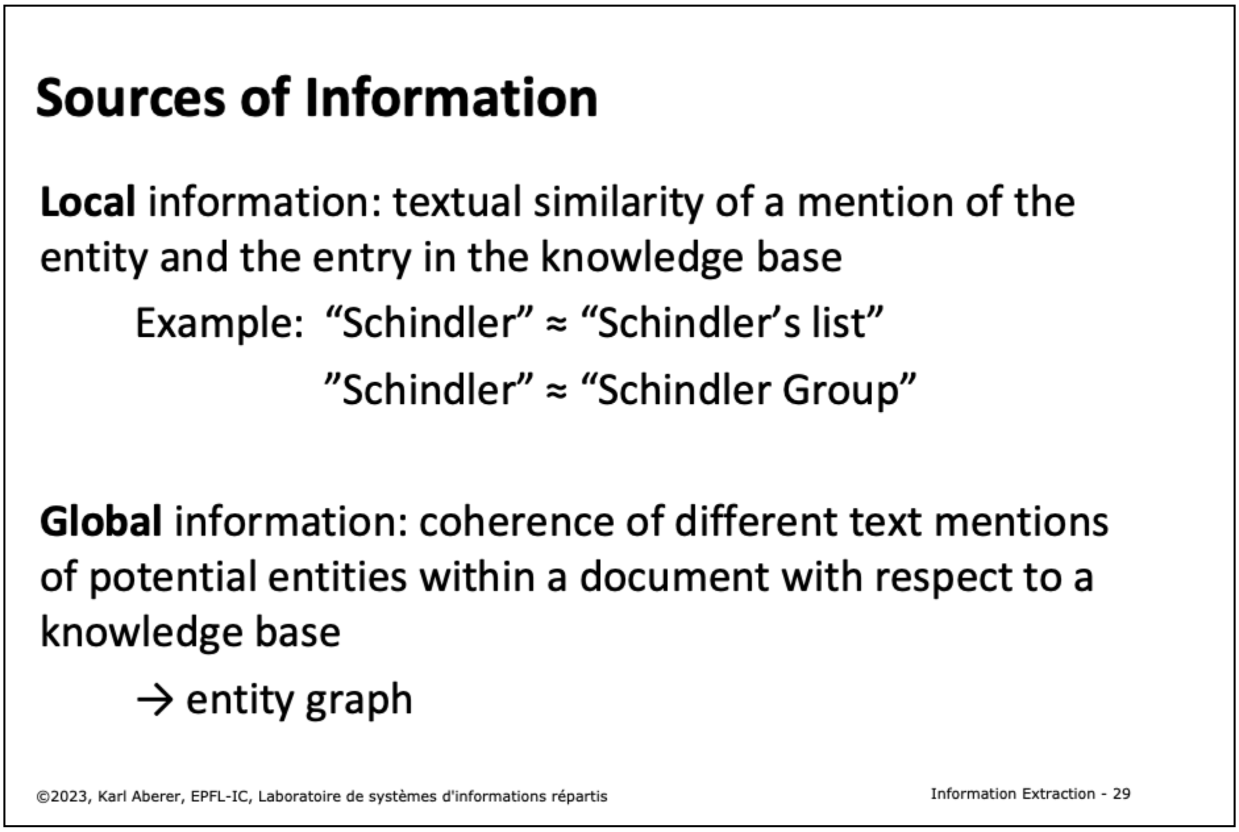
1 History
2 Geography
3 Demographics
4 Climate
5 See also
6 References

Town of Otis, Colorado

Town

Entering Otis from the east.

Entity disambiguation can however be quite challenging due to homonymy and synonymy. Handling these problems is essential for every text analytics tasks. Not being able to handle homonymy usually results in the introduction of noise into the results (poor precision), whereas not properly handling synonymy risks to miss relevant documents (poor recall).



Sources of Information

Local information: textual similarity of a mention of the entity and the entry in the knowledge base

Example: “Schindler” ≈ “Schindler’s list”
”Schindler” ≈ “Schindler Group”

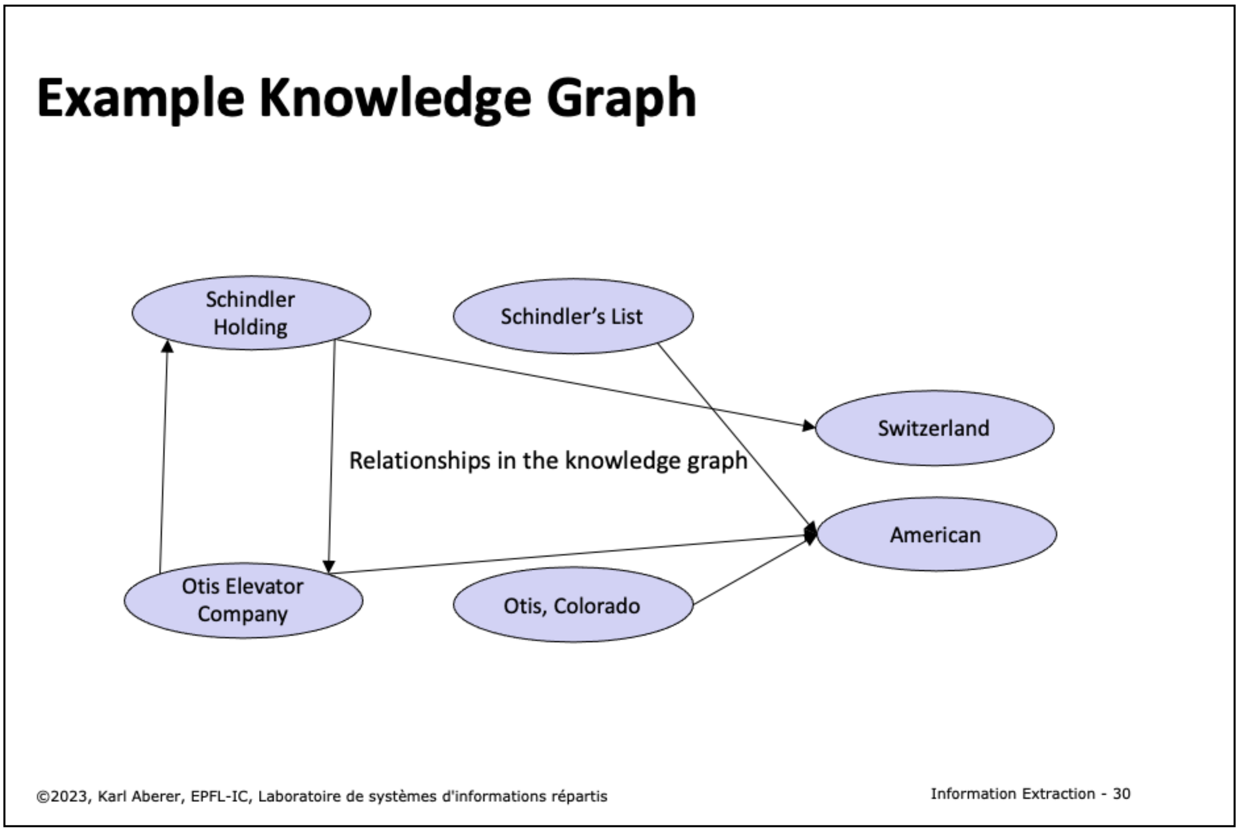
Global information: coherence of different text mentions of potential entities within a document with respect to a knowledge base

→ entity graph

For performing entity disambiguation one can exploit two different sources of information.

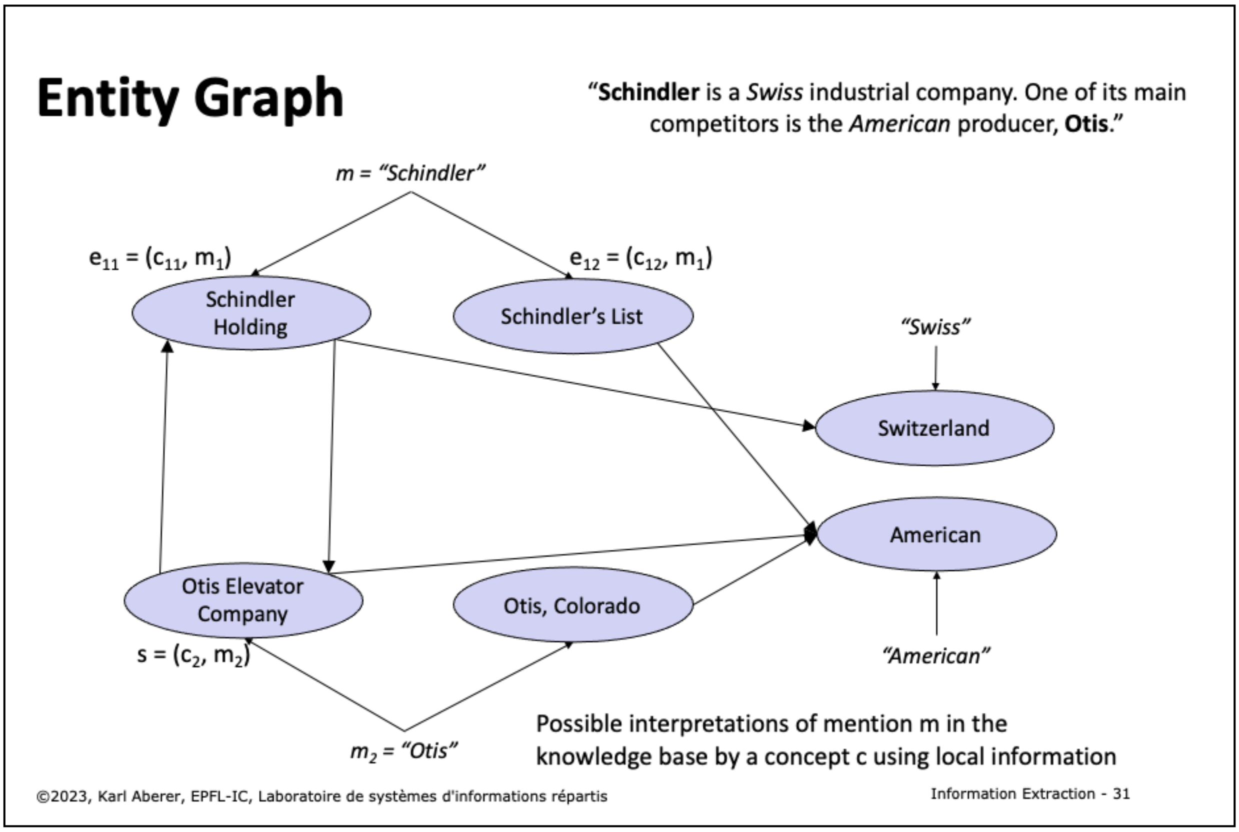
1. Local information extracted from the text mention, or its vicinity. This can be used to compare the text mention and its features with the text entry in the knowledge base, in order to obtain evidence which entities in the knowledge base are potential matches.

2. Coherence of different text mentions and knowledge base entries. When multiple entities are extracted from text, they will have relationships among each other. By analyzing whether the relationships among entries in the text and in the knowledge base are consistent with which each other one can disambiguate mentions of entities in the text.



Example Knowledge Graph

We will give now an example to illustrate the process of entity disambiguation using a knowledge graph. Assume the knowledge base contains the subgraph shown in the figure, with entities that apparently are relevant to our text example.



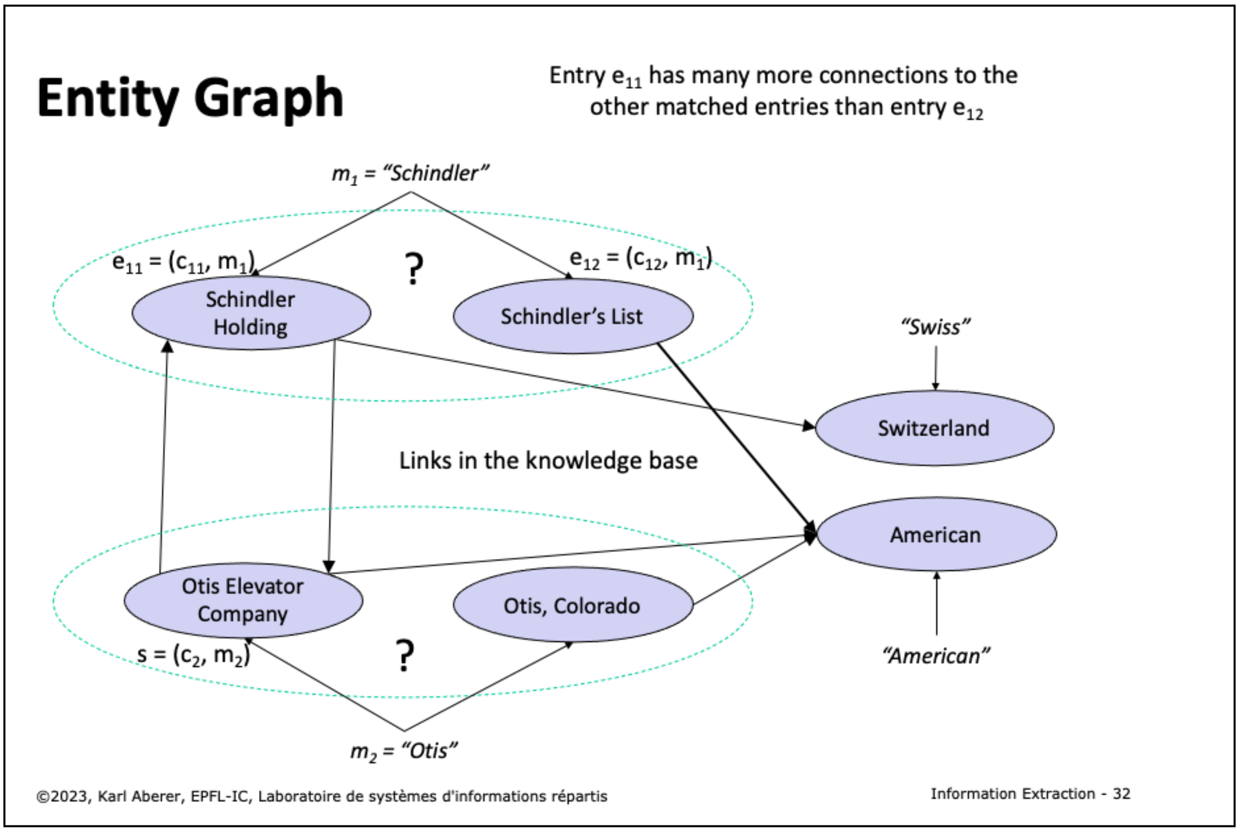
Entity Graph

"**Schindler** is a *Swiss* industrial company. One of its main competitors is the *American* producer, **Otis**."

Possible interpretations of mention m in the knowledge base by a concept c using local information

Information Extraction - 31

In a first step the textual mentions of the entities are linked to entries in the knowledge graph. After performing NER, this can be done using local information, textual similarity between the text mention and the name of the concept in the knowledge graph. By linking the text mentions we create entity matches of the form e = (c, m), where e is a node in the entity graph, c is the concept from the knowledge graph, and m is the textual mention of an entity in the text. The entity graph consists of all matched nodes in the knowledge graph, and the relationships among them. We may also associate a similarity measure to each node in the entity graph, capturing how well the text mention matches the concept in the knowledge graph.



Entity Graph

Entry e_{11} has many more connections to the other matched entries than entry e_{12}

m_1 = *"Schindler"*

$e_{11} = (c_{11}, m_1)$

?

$e_{12} = (c_{12}, m_1)$

Schindler Holding

Schindler's List

"Swiss"

Switzerland

Links in the knowledge base

American

Otis Elevator Company

Otis, Colorado

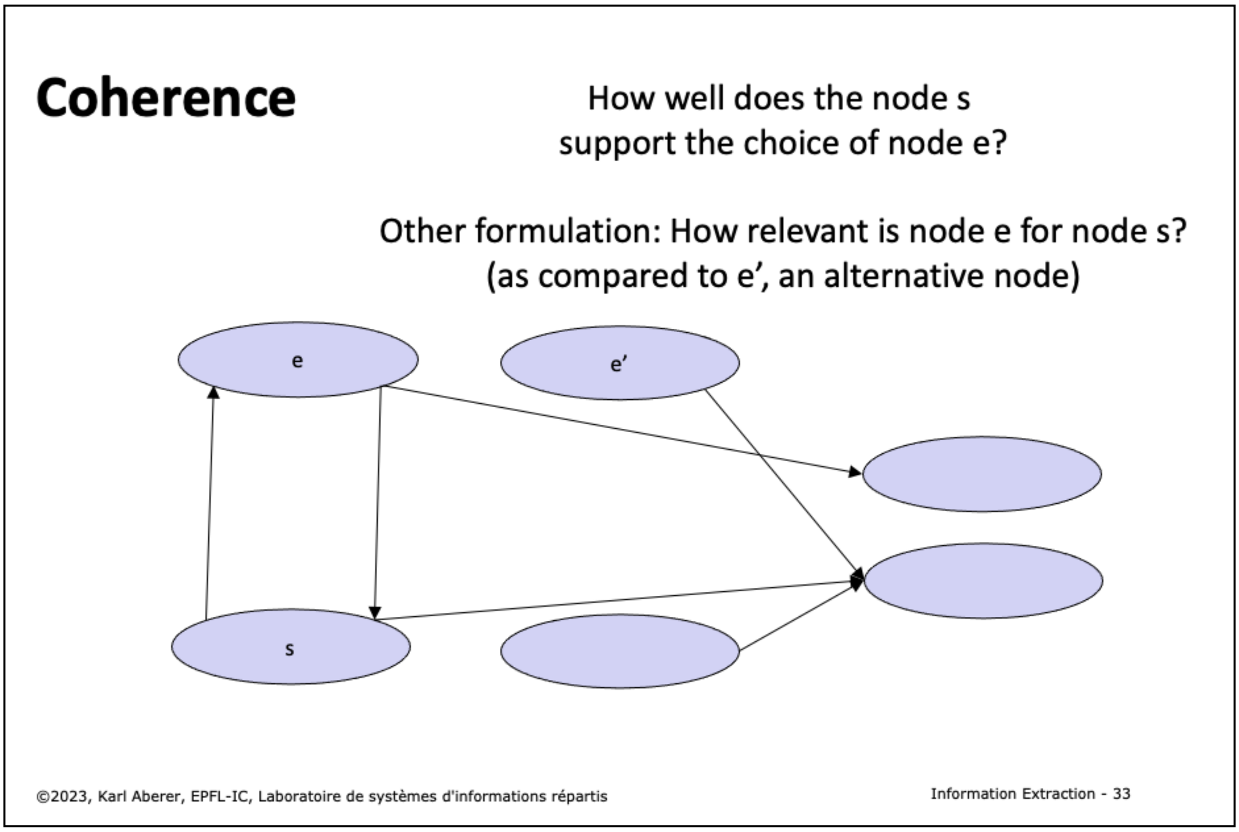
$s = (c_2, m_2)$

?

"American"

m_2 = *"Otis"*

The example shows that different interpretations are possible for both the mention "Schindler" and "Otis". Therefore, the problem is to decide which if each these two alternative matches is the better one. A possibility indication for the quality of a match is the connectivity of the nodes with the other nodes in the entity graph.



Coherence

How well does the node s support the choice of node e?

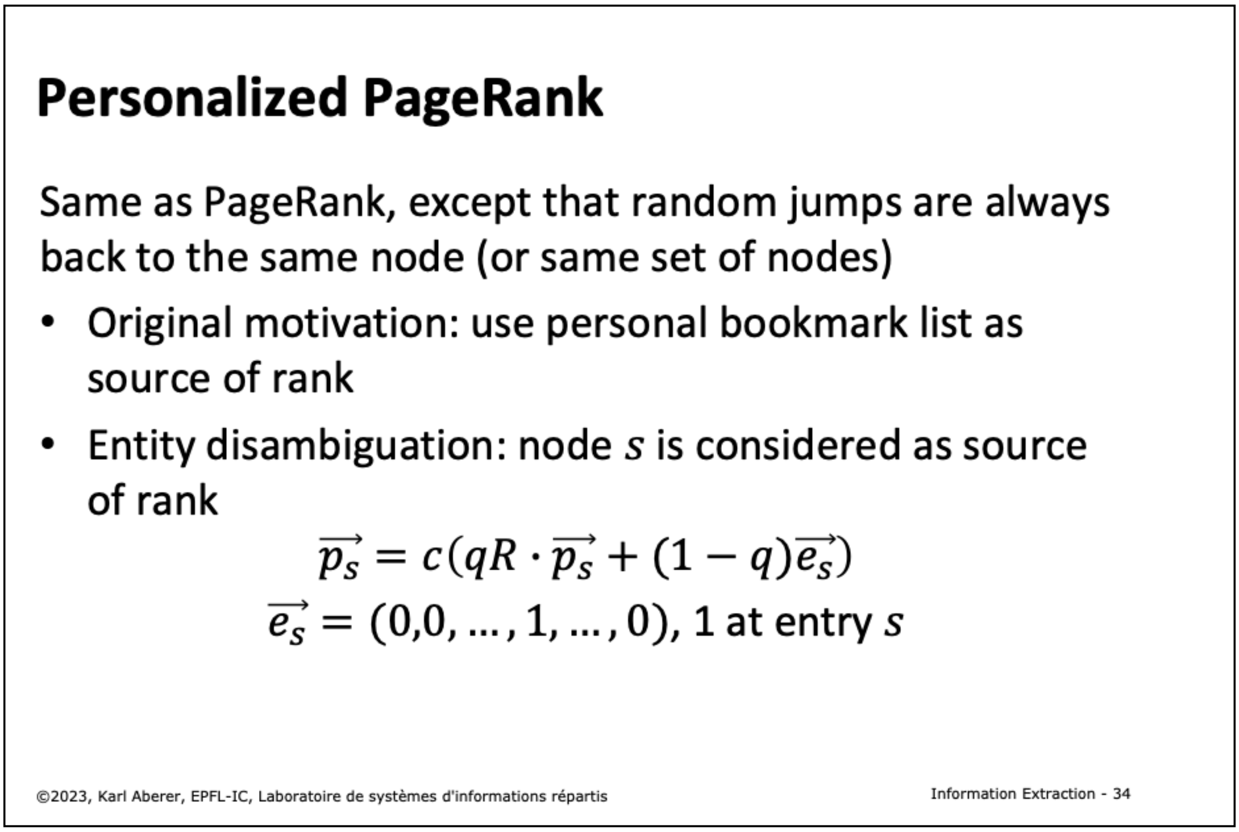
Other formulation: How relevant is node e for node s? (as compared to e’, an alternative node)

©2023, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Information Extraction - 33

Abstracting from the details of the example before, we can highlight the problem we need to solve. Given two alternative interpretations of a text mention, e and e’, and another node s in the entity graph, how well does the node s support the two alternatives. In the example graph we see that intuitively node s is much better connected with node e which indicates that e might indeed be the better choice.

The problem described bears similarity with another problem that has been addressed in the context of personalized Web search. Imagine the nodes are Web pages, s is a page that has been bookmarked by a user, and the question is which other pages, like e and e’, are also of interest to the user. From the example, it appears evident that it is page e. For making this intuition on connectedness operational, a variant of the PageRank algorithm has been proposed, called Personalized PageRank. The same algorithm has also been used to solve entity disambiguation.



Personalized PageRank

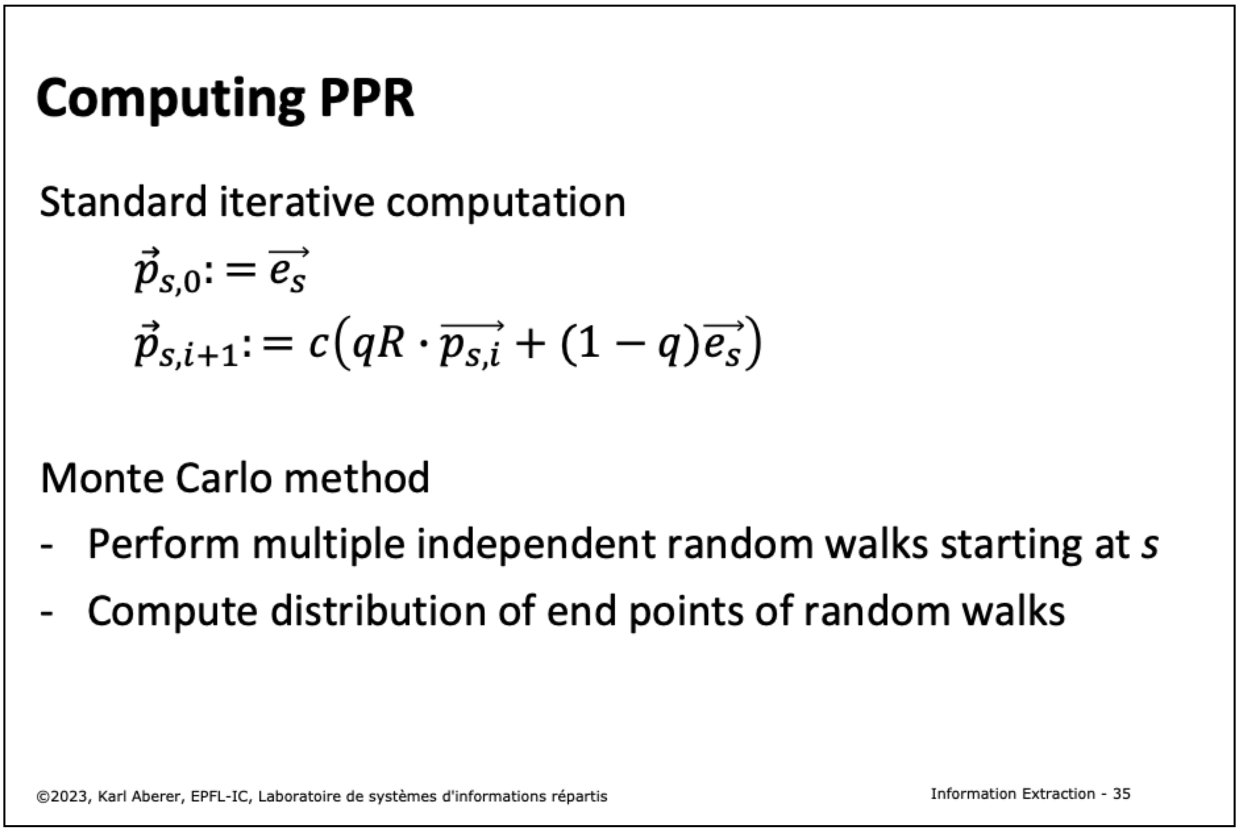
Same as PageRank, except that random jumps are always back to the same node (or same set of nodes)

- Original motivation: use personal bookmark list as source of rank
- Entity disambiguation: node s is considered as source of rank

$$\overrightarrow{p_s} = c(qR \cdot \overrightarrow{p_s} + (1-q)\overrightarrow{e_s})$$

$$\overrightarrow{e_s} = (0,0,\dots,1,\dots,0), \text{1 at entry } s$$

Personalized PageRank works almost the same as the original PageRank algorithm. The difference is that random jumps are not performed uniformly at random to nodes, but to a selected subset of nodes. In the context of personalized Web search this subset would be the personal bookmark list. By jumping back to the nodes from that list, it will have a large influence on the ranking of a page, such that nodes that are well connected to the bookmarks will receive a larger ranking value. In the context of entity disambiguation, the source is a selected node in the entity graph, and personalized page rank is used to compute how well other nodes in the entity graph are supported by that node.



Computing PPR

Standard iterative computation

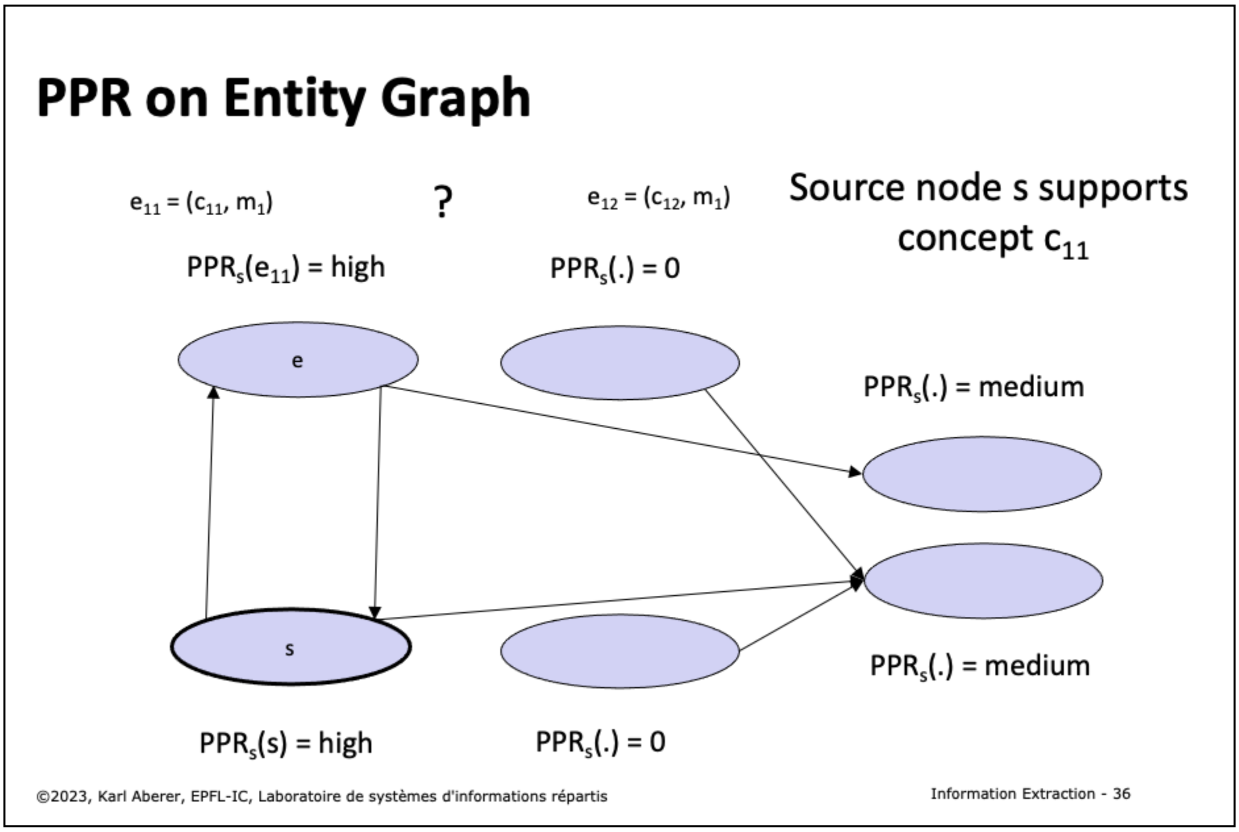
$$\vec{p}_{s,0} := \overrightarrow{e_s}$$

$$\vec{p}_{s,i+1} := c\left(qR \cdot \overrightarrow{p_{s,i}} + (1-q)\overrightarrow{e_s}\right)$$

Monte Carlo method

- Perform multiple independent random walks starting at *s*
- Compute distribution of end points of random walks

PPR can be computed either iteratively, like standard PageRank, or using a Monte Carlo method, by starting random walks independently at s and aggregating the distribution of the end points of those walks.



PPR on Entity Graph

$e_{11} = (c_{11}, m_1)$? $e_{12} = (c_{12}, m_1)$

Source node s supports concept c_{11}

$PPR_s(e_{11})$ = high

$PPR_s(.) = 0$

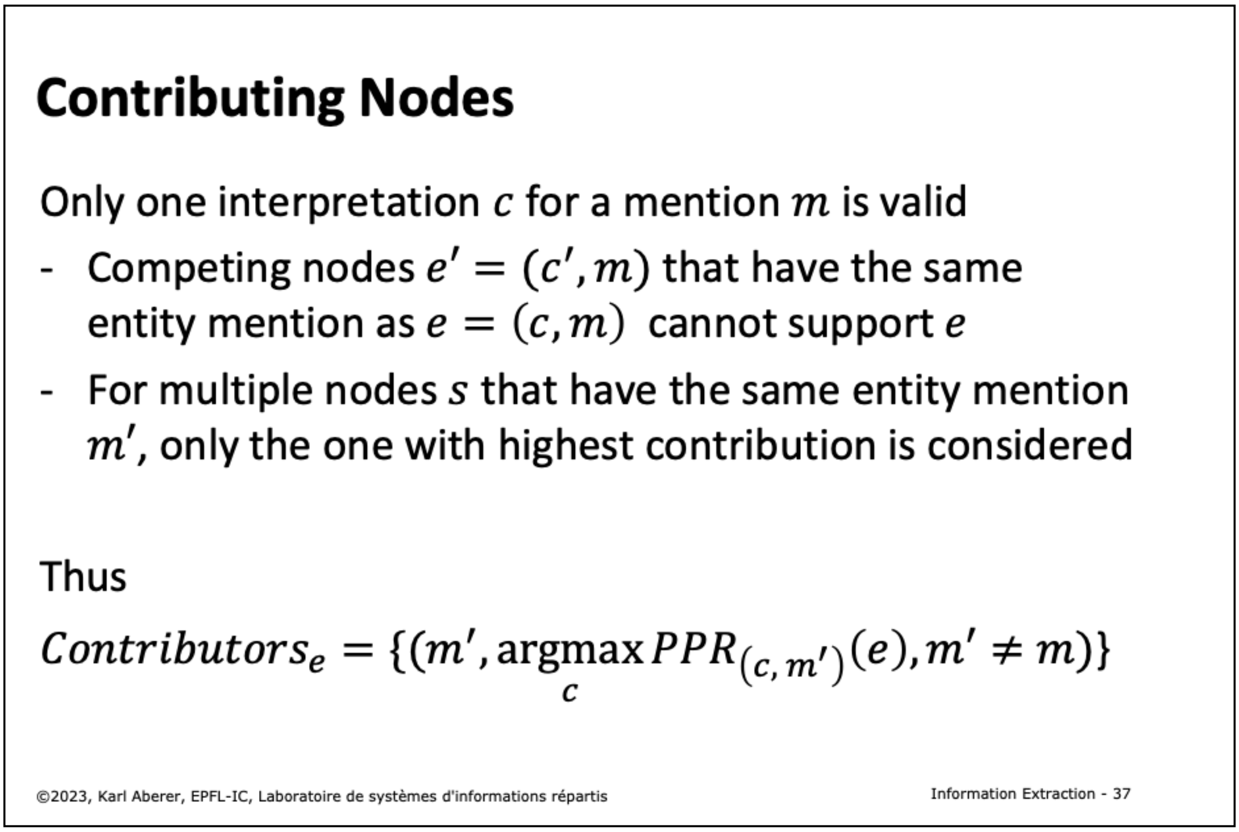
$PPR_s(.)$ = medium

$PPR_s(.)$ = medium

$PPR_s(s)$ = high

$PPR_s(.) = 0$

Applying PPR to the entity group, by considering a **source node s** as the source of rank, will generate a distribution of ranking values for all other nodes. Nodes that are well connected to s will receive higher ranking values. Intuitively it is clear, that in our example node e will be receive higher ranking when starting from s, and thus is the preferred interpretation for the entity matching mention m_1.



Contributing Nodes

Only one interpretation c for a mention m is valid

- Competing nodes $e' = (c', m)$ that have the same entity mention as $e = (c, m)$ cannot support e
- For multiple nodes s that have the same entity mention m', only the one with highest contribution is considered

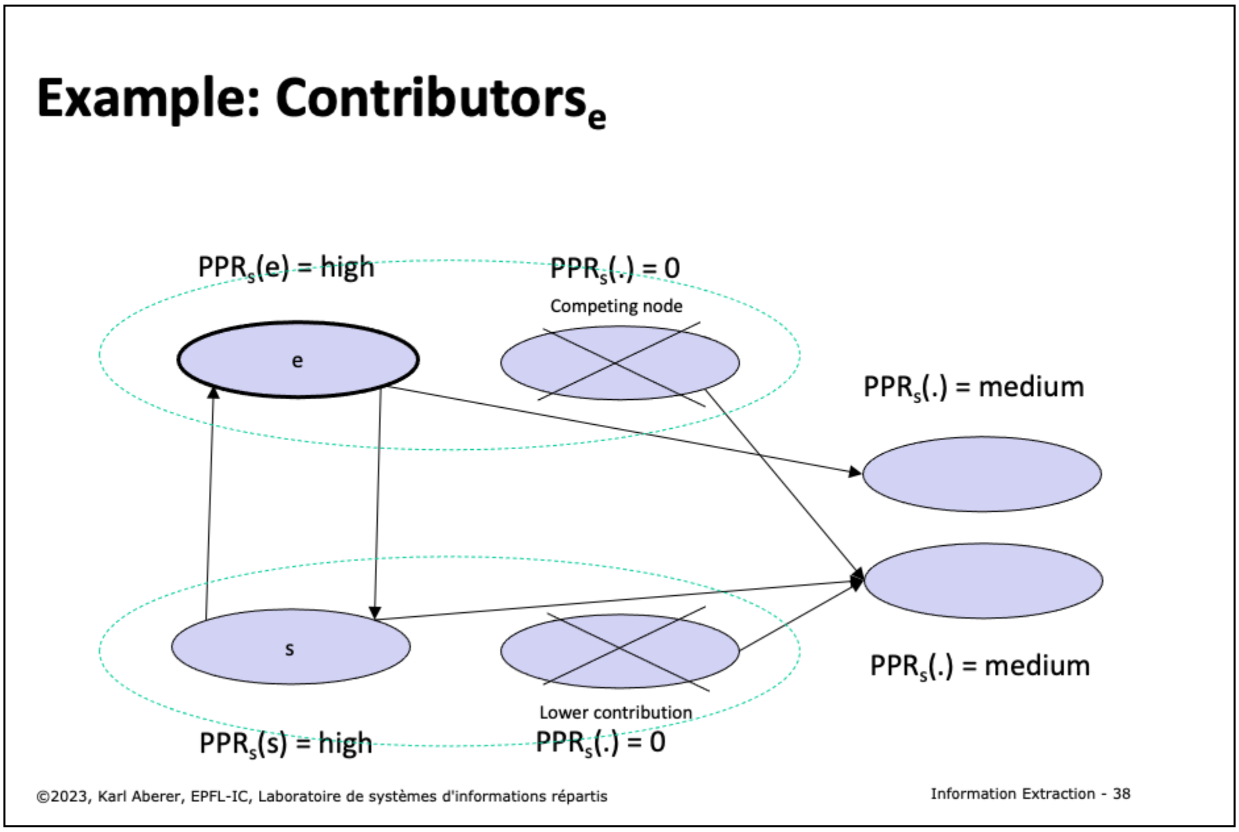
Thus

$$Contributors_e = \{(m', \underset{c}{\operatorname{argmax}} PPR_{(c,m')}(e), m' \neq m)\}$$

The question is which nodes s should contribute to the disambiguation of a mention m. The considerations for choosing those nodes take into account the following two issues:

- When we are computing the support for an interpretation $e = (c, m)$ of text mention m, with competing interpretations $e' = (c', m)$, the competing node should not be used to produce support for e. So, these nodes are excluded.
- When there exist multiple nodes for a text mention m' , only the one that is producing the largest contribution to the interpretation $e = (c, m)$ of text mention m is considered. This makes sense, since the other nodes related to m' might favor an alternative interpretation for m, and therefore should not be considered.

This results in a set of contributing nodes $Contributors_e$ for each node $e = (c, m)$ in the entity graph.



Example: Contributors$_e$

$PPR_s(e)$ = high

$PPR_s(.)$ = 0

Competing node

e

$PPR_s(.)$ = medium

s

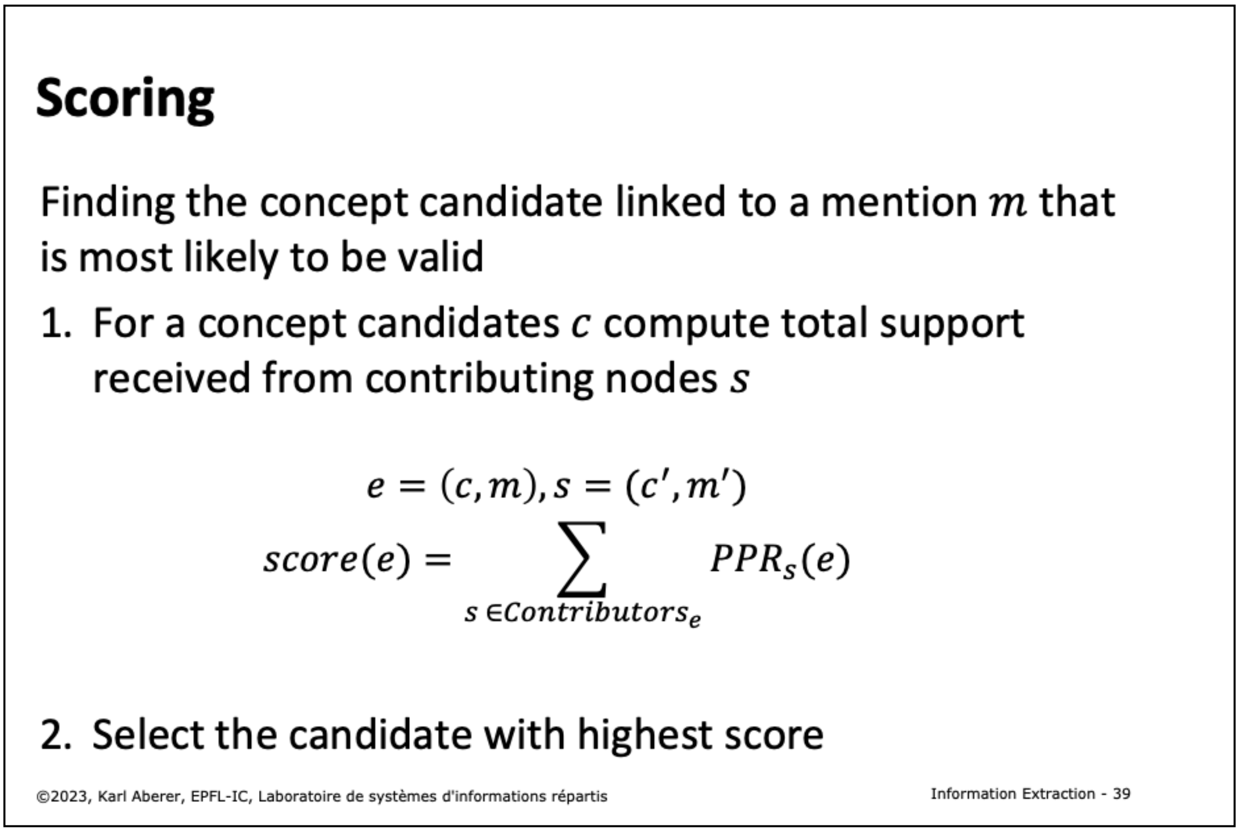
$PPR_s(.)$ = medium

Lower contribution

$PPR_s(s)$ = high

$PPR_s(.)$ = 0

This figure shows that for the computation of the score of node e using node s as source, two nodes from the entity graph will be excluded. First, the node that is linked to the same text mention as e and is a competing node, and second the node that is linked to the same text mention and s and is producing a lower score.



Scoring

Finding the concept candidate linked to a mention m that is most likely to be valid

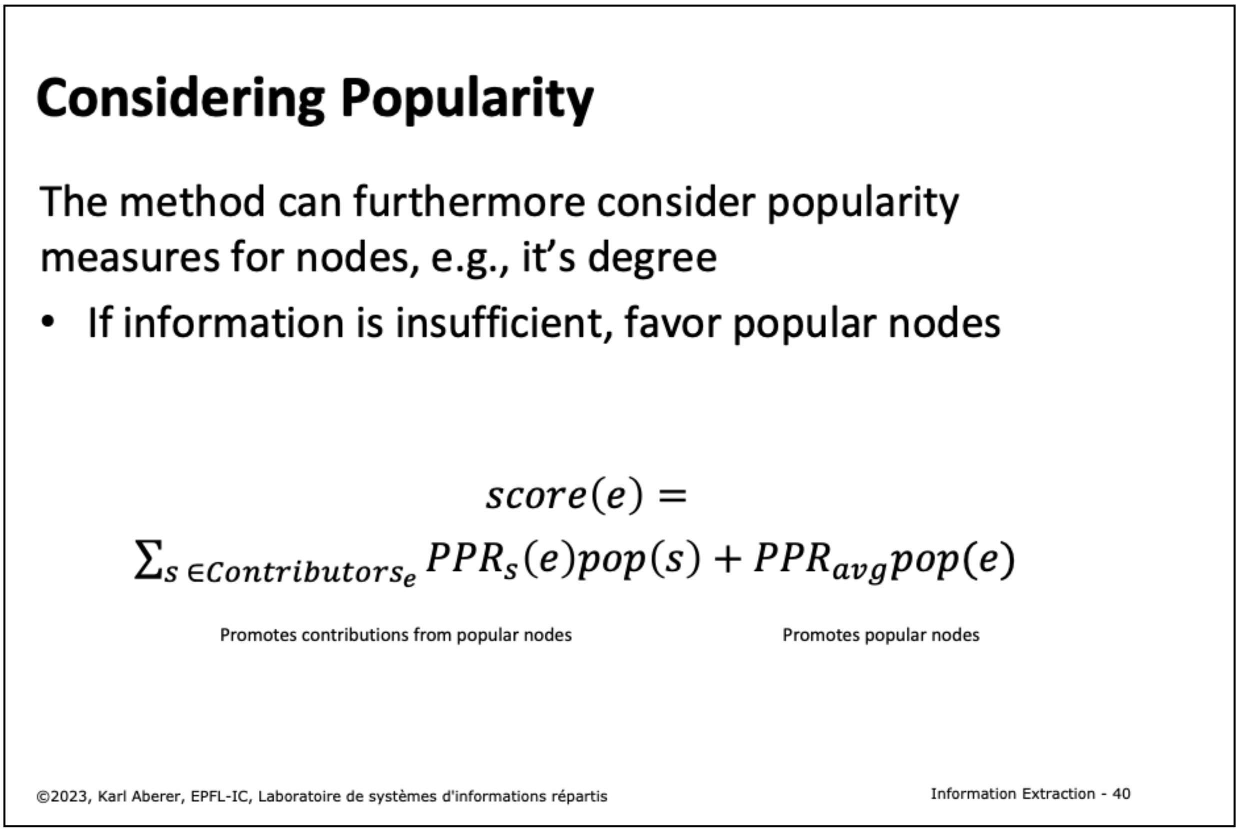
1. For a concept candidates c compute total support received from contributing nodes s

$$e = (c, m), s = (c', m')$$

$$score(e) = \sum_{s \in Contributors_e} PPR_s(e)$$

2. Select the candidate with highest score

Using the contributing nodes, the personalized PageRank scores that they contribute are added and the candidate with the best score is selected.



Considering Popularity

The method can furthermore consider popularity measures for nodes, e.g., it’s degree

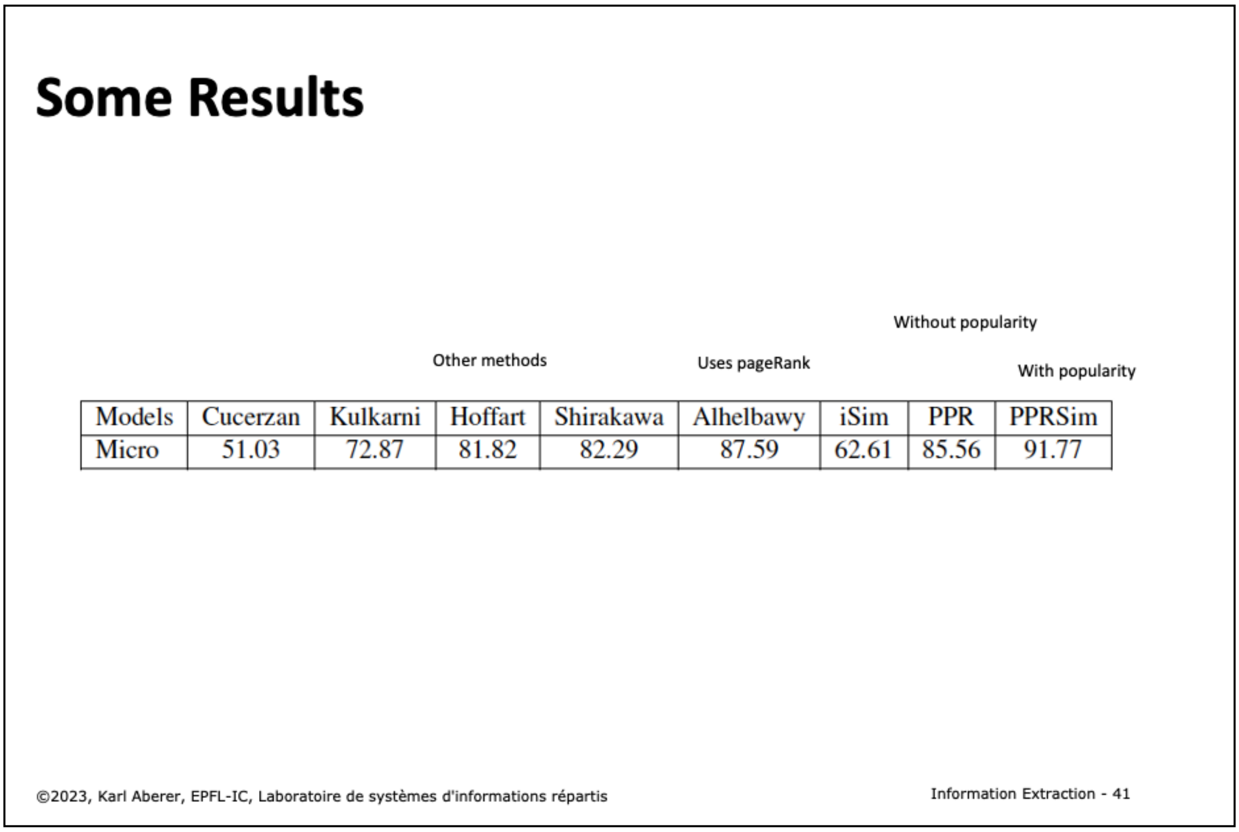
- If information is insufficient, favor popular nodes

$$score(e) = \sum_{s \in Contributors_e} PPR_s(e)pop(s) + PPR_{avg}pop(e)$$

Promotes contributions from popular nodes

Promotes popular nodes

To further improve the method, it is possible to add a general popularity measure as a weight the contributions of source nodes. This will favor the contribution of popular nodes, which is beneficial if little information is available for disambiguation. In such a case, it is better to choose a popular candidate since chances that an interpretation supported by a popular node are higher. One of possible choice of a popularity score could be the number of links a node has in the knowledge base. Similarly, also the popularity of the candidates e can be used as a contribution to the score.



Some Results

Without popularity

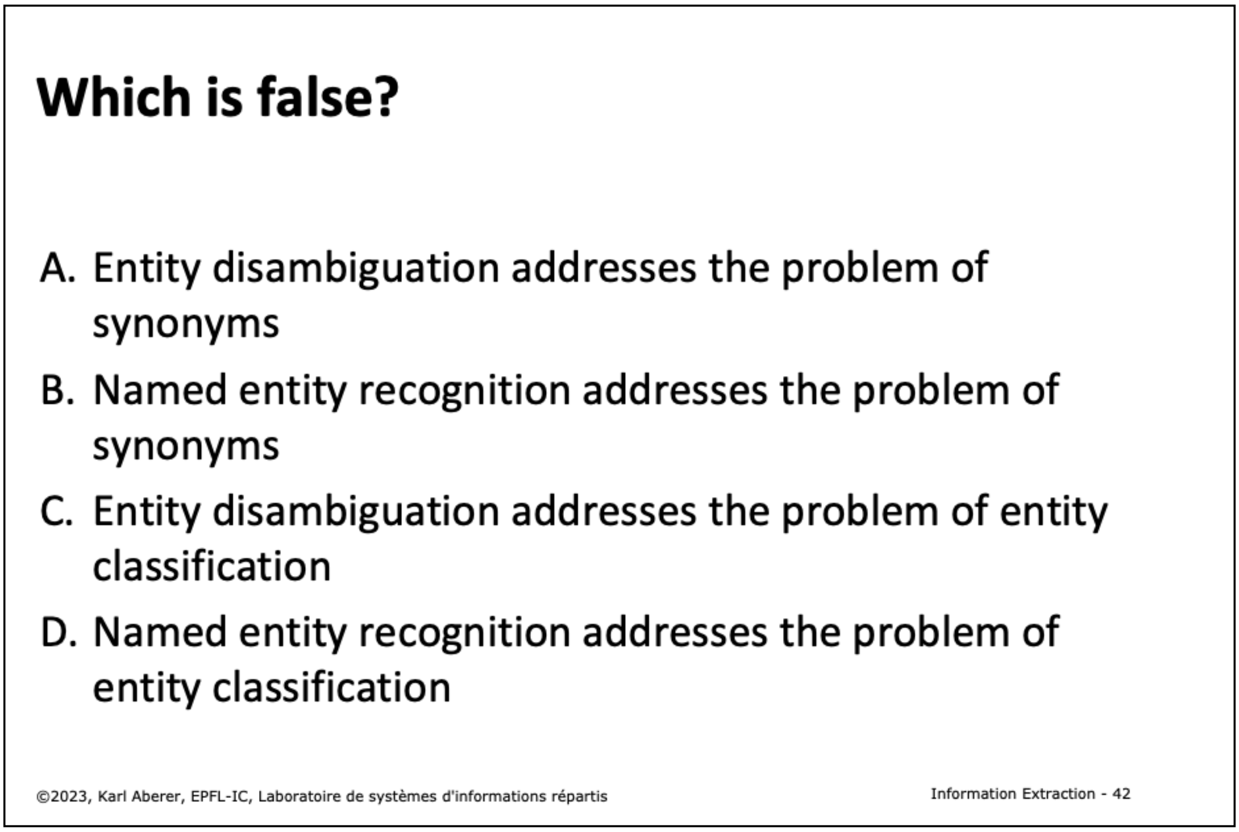
Other methods

Uses pageRank

With popularity

Models	Cucerzan	Kulkarni	Hoffart	Shirakawa	Alhelbawy	iSim	PPR	PPRSim
Micro	51.03	72.87	81.82	82.29	87.59	62.61	85.56	91.77

Experimental results show that the method works relatively well, with around 90% of entities that are correctly disambiguated. One can observe that the use of popularity helps to slightly improve the results.



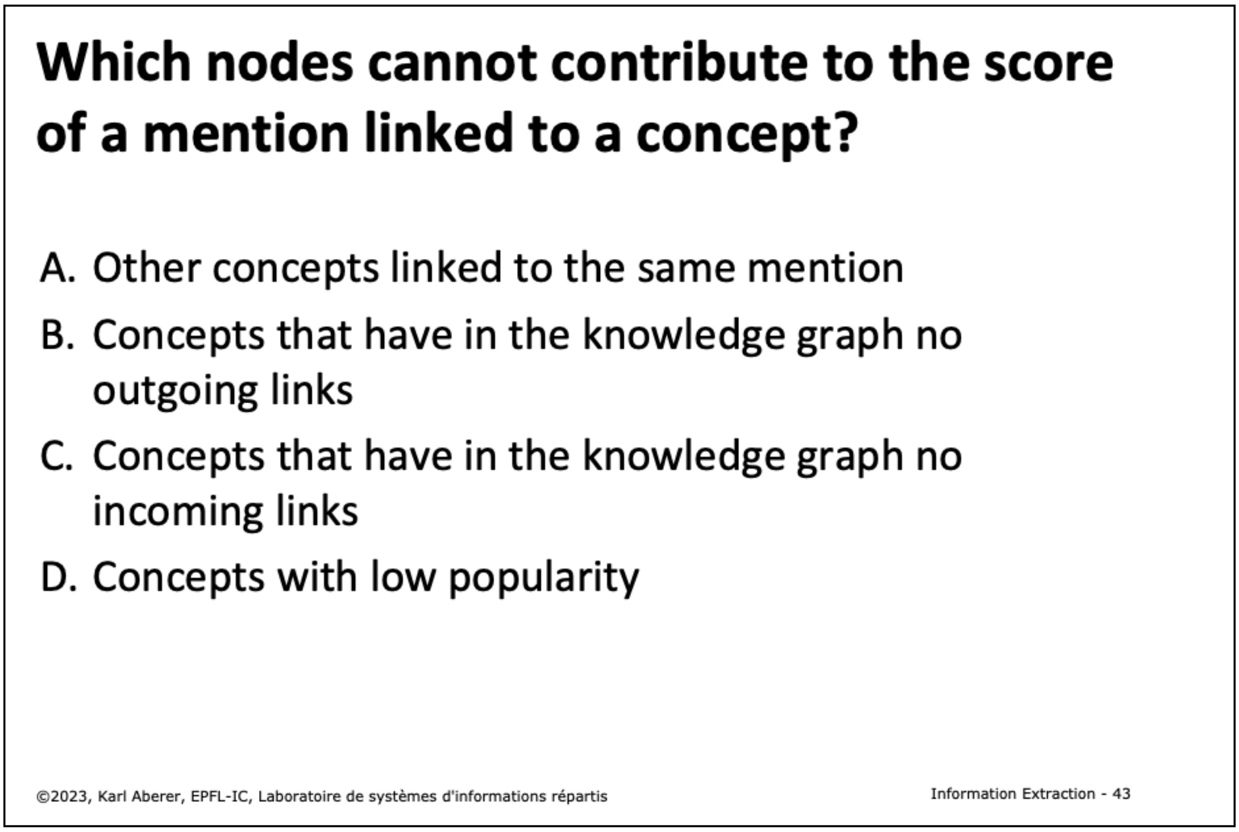
Which is false?

A. Entity disambiguation addresses the problem of synonyms

B. Named entity recognition addresses the problem of synonyms

C. Entity disambiguation addresses the problem of entity classification

D. Named entity recognition addresses the problem of entity classification



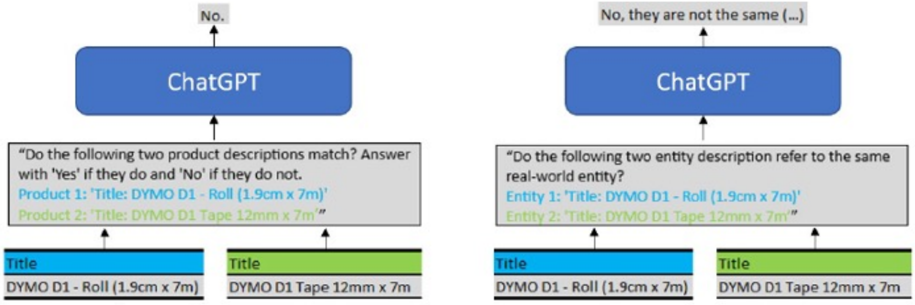
Which nodes cannot contribute to the score of a mention linked to a concept?

A. Other concepts linked to the same mention
B. Concepts that have in the knowledge graph no outgoing links
C. Concepts that have in the knowledge graph no incoming links
D. Concepts with low popularity

4.1.4 Entity Matching with GPT

GPT can be prompted to directly decide whether to entities are the same

- the challenge is to design good prompts



"Do the following two product descriptions match? Answer with 'Yes' if they do and 'No' if they do not.
Product 1: 'Title: DYMO D1 - Roll (1.9cm x 7m)'
Product 2: 'Title: DYMO D1 Tape 12mm x 7m'"

"Do the following two entity description refer to the same real-world entity?
Entity 1: 'Title: DYMO D1 - Roll (1.9cm x 7m)'
Entity 2: 'Title: DYMO D1 Tape 12mm x 7m'"

No.
ChatGPT
Title
DYMO D1 - Roll (1.9cm x 7m)
Title
DYMO D1 Tape 12mm x 7m
No, they are not the same (...)
ChatGPT
Title
DYMO D1 - Roll (1.9cm x 7m)
Title
DYMO D1 Tape 12mm x 7m

Considerations in Prompt Design

General vs. domain-specific

- “are these entities the same?” vs. “are these products the same”

Simple vs. complex

- ”match” vs. “refer to the same real-world entity”

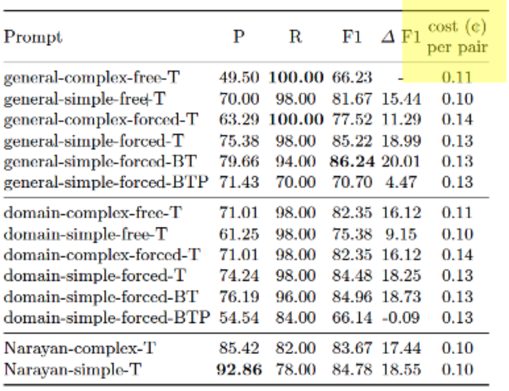
Free vs. forced answer

- “answers with ‘yes’ or ‘no” vs. no instruction

Adding entity attributes

- Adding price information or not

Sample evaluation results



Prompt	P	R	F1	Δ F1	cost (c) per pair
general-complex-free-T	49.50	**100.00**	66.23	-	0.11
general-simple-free-T	70.00	98.00	81.67	15.44	0.10
general-complex-forced-T	63.29	**100.00**	77.52	11.29	0.14
general-simple-forced-T	75.38	98.00	85.22	18.99	0.13
general-simple-forced-BT	79.66	94.00	**86.24**	20.01	0.13
general-simple-forced-BTP	71.43	70.00	70.70	4.47	0.13
domain-complex-free-T	71.01	98.00	82.35	16.12	0.11
domain-simple-free-T	61.25	98.00	75.38	9.15	0.10
domain-complex-forced-T	71.01	98.00	82.35	16.12	0.14
domain-simple-forced-T	74.24	98.00	84.48	18.25	0.13
domain-simple-forced-BT	76.19	96.00	84.96	18.73	0.13
domain-simple-forced-BTP	54.54	84.00	66.14	-0.09	0.13
Narayan-complex-T	85.42	82.00	83.67	17.44	0.10
Narayan-simple-T	**92.86**	78.00	84.78	18.55	0.10

Findings on what helps
- domain-specific wording
- simpler wording
- forcing an answer

Findings on what hurts
- price information or products (lack of normalization)

In-context Learning

Providing sample training data

Matches:

Product 1: 'Title: SANDISK EXTREME PRO SDHC 32GB 300MB/S UHS-II U3' Product 2: 'Title: Sandisk SDXC card Extreme Pro UHS-II, 32gb, 300mbps'

Product 1: 'Title: Dymo 53718 Black On Yellow - 24mm' Product 2: 'Title: Dymo 24mm Black On Yellow D1 Tape (53718)'

Product 1: 'Title: DS-7216HQHI-K1 Hikvision 16 cs. TurboHD DVR' Product 2: 'Title: Hikvision DS-7216HQHI-K1 Turbo HD DVR'

Product 1: 'Title: APCRBC133APC Replacement Battery Cartridge #133' Product 2: 'Title: APC RBC133 Replacement Battery Cartridge'

Product 1: 'Title: Gigabyte NVIDIA GeForce GTX 1650 4GB D6 OC Turing Graphics Card' Product 2: 'Title: GigaByte GeForce GTX 1650 D6 Windforce OC 4G'

Non-matches:

Product 1: 'Title: RAM Corsair ValueSelect DDR4 2133MHz 1x8Go' Product 2: 'Title: CORSAIR New 8gb (1x8gb) Ddr4 2666mhz Vengeance CMK8GX4M1A2666C16'

Product 1: 'Title: Evolis Zenius/Primacy Black Monochrome Ribbon 2000 image RCT023NAA' Product 2: 'Title: Zebra 800015-101 Black Monochrome Ribbon - 1000 Prints'

Product 1: 'Title: 128GB Pendrive SanDisk Extreme PRO SSD USB 3.1 420MB/s' Product 2: 'Title: SanDisk 128GB Extreme Pro SDXC 150MB/s V-30 UHS-1 U3 Memory Card'

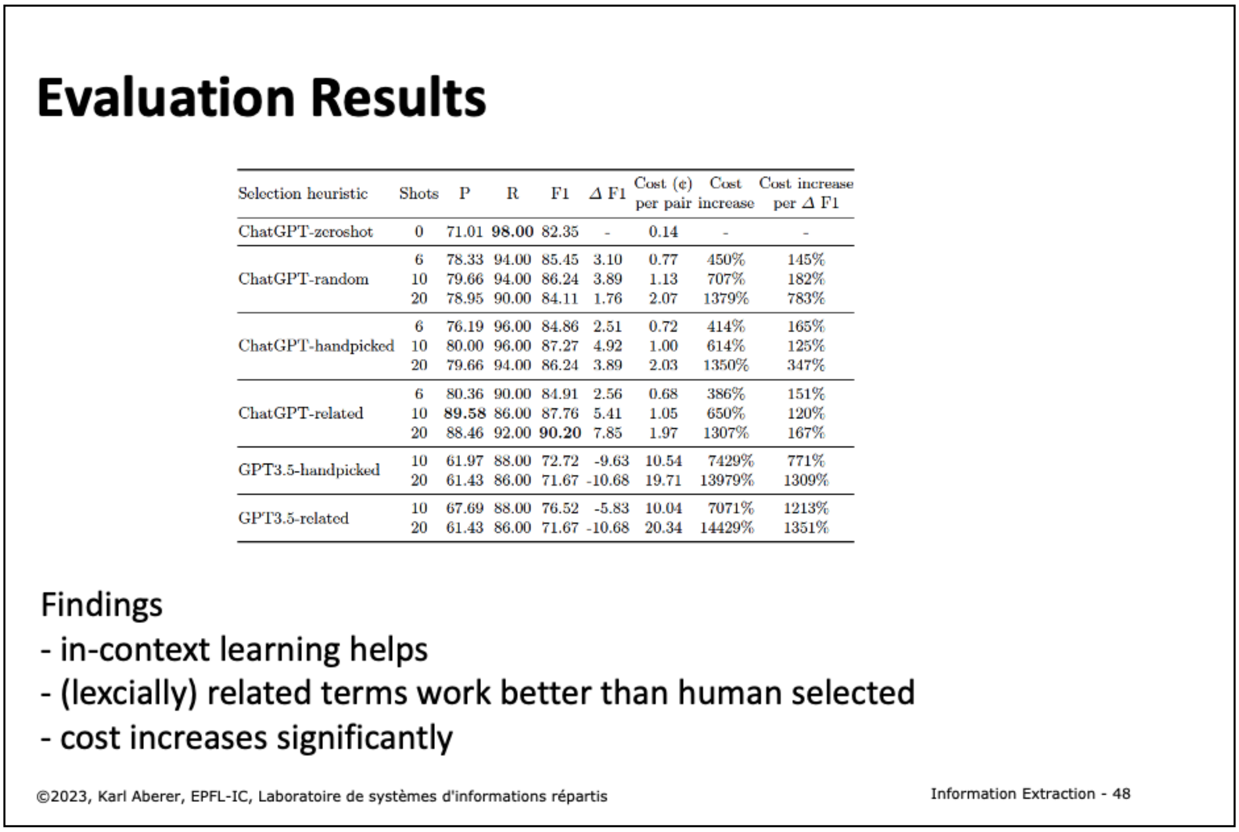
Product 1: 'Title: Samsung SSD 970 EVO Plus 250GB' Product 2: 'Title: Samsung 970 EVO SSD M.2 2280 - 500GB'

Product 1: 'Title: Akumulator APC Replacement Battery Cartridge #110' Product 2: 'Title: APC - Replacement Battery Cartridge #117'

Do the following two product descriptions refer to the same real-world product? Answer with 'Yes' if they do and 'No' if they do not.

Product 1: 'Title: MEMORIA SD SANDISK ULTRA SDHC SDXC UHS I CARD 100 MBS SDSDUNR 256G GN6IN'

Product 2: 'Title: SanDisk Extreme Pro (256GB) SDXC Memory Card'



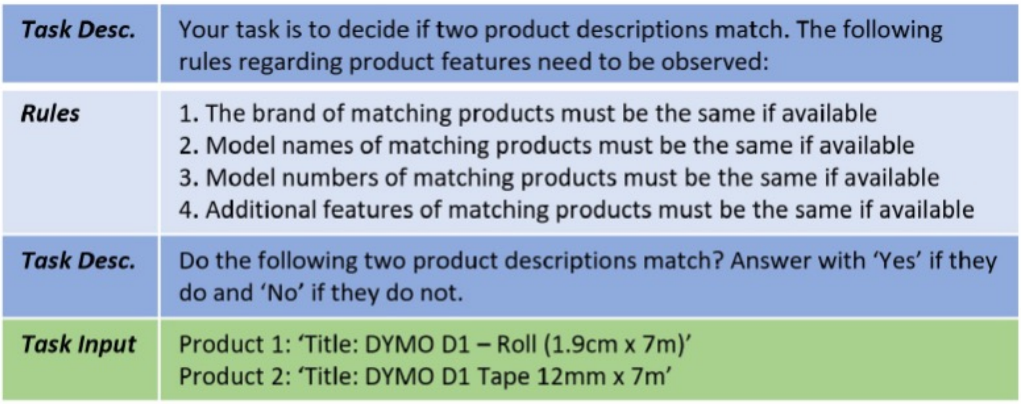
Evaluation Results

Selection heuristic	Shots	P	R	F1	Δ F1	Cost (¢) per pair	Cost increase	Cost increase per Δ F1
ChatGPT-zeroshot	0	71.01	**98.00**	82.35	-	0.14	-	-
ChatGPT-random	6	78.33	94.00	85.45	3.10	0.77	450%	145%
	10	79.66	94.00	86.24	3.89	1.13	707%	182%
	20	78.95	90.00	84.11	1.76	2.07	1379%	783%
ChatGPT-handpicked	6	76.19	96.00	84.86	2.51	0.72	414%	165%
	10	80.00	96.00	87.27	4.92	1.00	614%	125%
	20	79.66	94.00	86.24	3.89	2.03	1350%	347%
ChatGPT-related	6	80.36	90.00	84.91	2.56	0.68	386%	151%
	10	**89.58**	86.00	87.76	5.41	1.05	650%	120%
	20	88.46	92.00	**90.20**	7.85	1.97	1307%	167%
GPT3.5-handpicked	10	61.97	88.00	72.72	-9.63	10.54	7429%	771%
	20	61.43	86.00	71.67	-10.68	19.71	13979%	1309%
GPT3.5-related	10	67.69	88.00	76.52	-5.83	10.04	7071%	1213%
	20	61.43	86.00	71.67	-10.68	20.34	14429%	1351%

Findings
- in-context learning helps
- (lexcially) related terms work better than human selected
- cost increases significantly

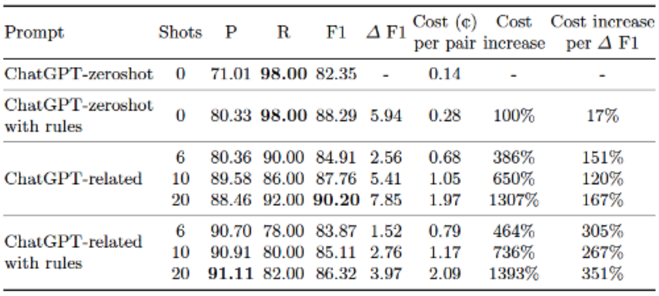
Lexically related = high Jaccard similarity

Providing Matching Rules



Task Desc.	Your task is to decide if two product descriptions match. The following rules regarding product features need to be observed:
Rules	1. The brand of matching products must be the same if available 2. Model names of matching products must be the same if available 3. Model numbers of matching products must be the same if available 4. Additional features of matching products must be the same if available
Task Desc.	Do the following two product descriptions match? Answer with 'Yes' if they do and 'No' if they do not.
Task Input	Product 1: 'Title: DYMO D1 – Roll (1.9cm x 7m)' Product 2: 'Title: DYMO D1 Tape 12mm x 7m'

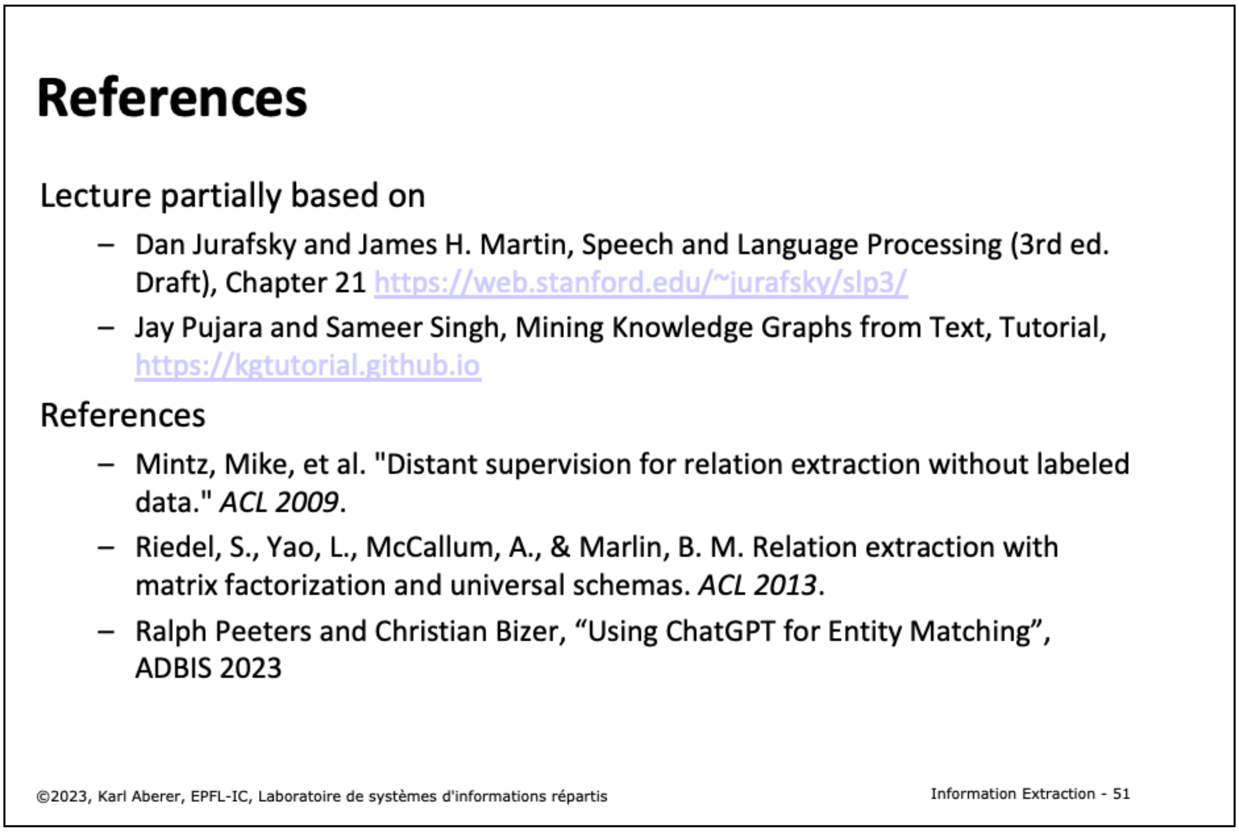
Evaluation Results



Prompt	Shots	P	R	F1	Δ F1	Cost (¢) per pair	Cost increase	Cost increase per Δ F1
ChatGPT-zeroshot	0	71.01	**98.00**	82.35	-	0.14	-	-
ChatGPT-zeroshot with rules	0	80.33	**98.00**	88.29	5.94	0.28	100%	17%
ChatGPT-related	6	80.36	90.00	84.91	2.56	0.68	386%	151%
	10	89.58	86.00	87.76	5.41	1.05	650%	120%
	20	88.46	92.00	**90.20**	7.85	1.97	1307%	167%
ChatGPT-related with rules	6	90.70	78.00	83.87	1.52	0.79	464%	305%
	10	90.91	80.00	85.11	2.76	1.17	736%	267%
	20	**91.11**	82.00	86.32	3.97	2.09	1393%	351%

Findings
- rules with in-context learning increases precision at cost of recall

Note that rules do not require training data



References

Lecture partially based on

- Dan Jurafsky and James H. Martin, Speech and Language Processing (3rd ed. Draft), Chapter 21 https://web.stanford.edu/~jurafsky/slp3/
- Jay Pujara and Sameer Singh, Mining Knowledge Graphs from Text, Tutorial, https://kgtutorial.github.io

References

- Mintz, Mike, et al. "Distant supervision for relation extraction without labeled data." *ACL 2009*.
- Riedel, S., Yao, L., McCallum, A., & Marlin, B. M. Relation extraction with matrix factorization and universal schemas. *ACL 2013*.
- Ralph Peeters and Christian Bizer, "Using ChatGPT for Entity Matching", ADBIS 2023