Part 3: Knowledge Modeling – Information Extraction

Populating Knowledge Bases

Manual creation of knowledge bases is expensive

Can we produce such knowledge 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. So an interesting problem is whether such knowledge bases could not be automatically constructed.

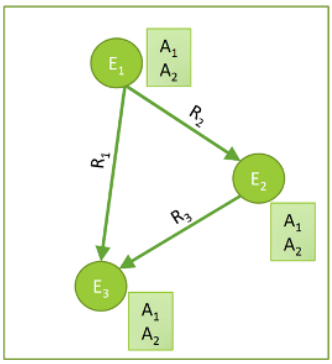
For automatic construction we can exploit all information that is digitally available, e.g., all documents accessible on the Web. These documents encode massive human knowledge in natural language. So the problem is to extract such knowledge by analyzing natural language text, which is not an easy problem.

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, in particular the documents from which the

knowledge has been extracted, for supporting more expressive and precise searches and analysis.

Knowledge Extraction

From text to knowledge



Who are the entities?

What are their attributes?

How are they related?

For extracting knowledge from textual content, we can consider the different constituents of a knowledge graph separately: entities, attributes and relationships.

Basic Questions in Knowledge Extraction

Who are the entities?

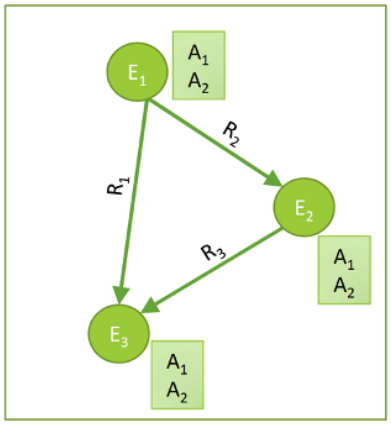
- Keyphrase extraction
- Named entity recognition

What are their attributes?

- Named entity recognition

How are they related?

- Relation (information) extraction
- Taxonomy Induction



For each of the three questions of generating the basic entities for a knowledge graph there exist specific problems that have been investigated.

Keyphrase extraction concerns the extraction of typical phrases in a document that could identify basic concepts.

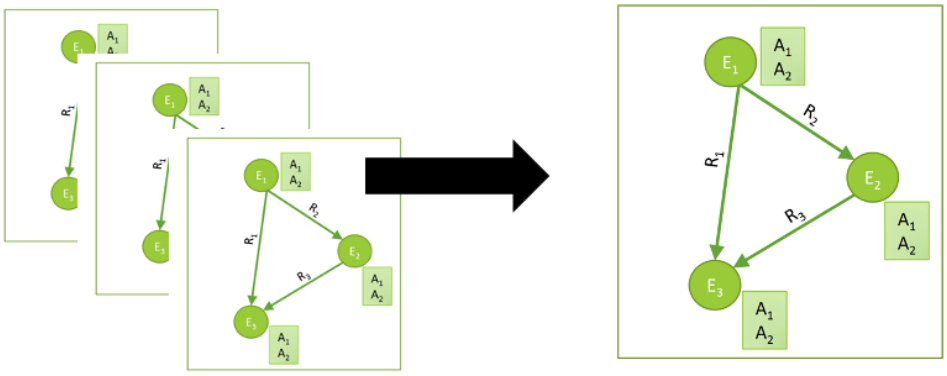
Named entity extaction concerns the extaction and typing of text that represents names of real-world entities.

Information extraction concerns the automated extraction of relationships from text.

Taxonomy induction concerns the automated extraction of a specific relationship, namely generalization, from text.

Knowledge Inference

From available knowledge to more complete and precise knowledge



Once knowledge graphs have been extracted from text they can be further processed. This enables the inference of new knowledge from the existing knowledge, but as well the correction, completion and integration of existing knowledge bases.

Basic Questions in Knowledge Inference

Who are the entities?

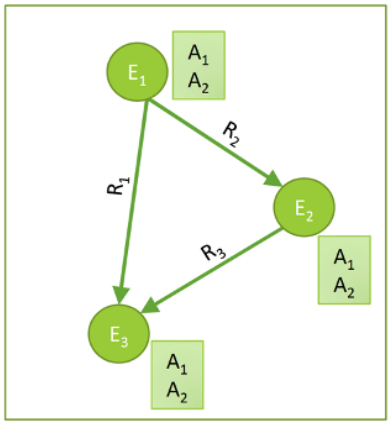
- ➢ Entity Linking / Disambiguation
- ➢ Schema integration

What are their attributes?

- ➢ Collective Classification

How are they related?

- ➢ Link prediction



Knowledge inference concerns a wide number of problems that have been studied in many different contexts. Some of the basic examples are:

Entity linking and disambiguation, which concerns the problem of identifying which entity names represent the same real-world entity, respective which entity is referred to in case of ambiguous entity names.

Schema integration, which concerns the problem which classes, attributes and relationships in one knowledge bases correspond to which in another one.

Collective classification, which concerns the problem of learning unknown attribute values from the available knowledge in a knowledge base.

Link prediction, which concerns the problem of learning unknown relationships from the available knowledge in a knowledge base.

Knowledge Extraction

1. Keyphrase extraction
2. Named entity recognition
3. Information extraction
4. Taxonomy Induction

Knowledge Inference

1. Entity Disambiguation
2. Label Propagation
3. Link Prediction
4. Data Integration

1. KEYPHRASE EXTRACTION

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 (phrase = sequence of words) that are particularly typical for the document and characterize important concepts that occur in the document. Key phrase extraction has been developed to support document summarization, where the key phrase give an overview of the key concepts of a document, and document search and indexing, where a distinctive vocabulary is being used in document searches. Moreover, key phrases can also provide useful features for document classification, i.e. they key phrase extraction can be considered as a feature selection method, and they support opinion mining, identifying distinctive expressions related to opinions. In the example text we see the possible outcome of key phrase extraction, with all key phrases identifies 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

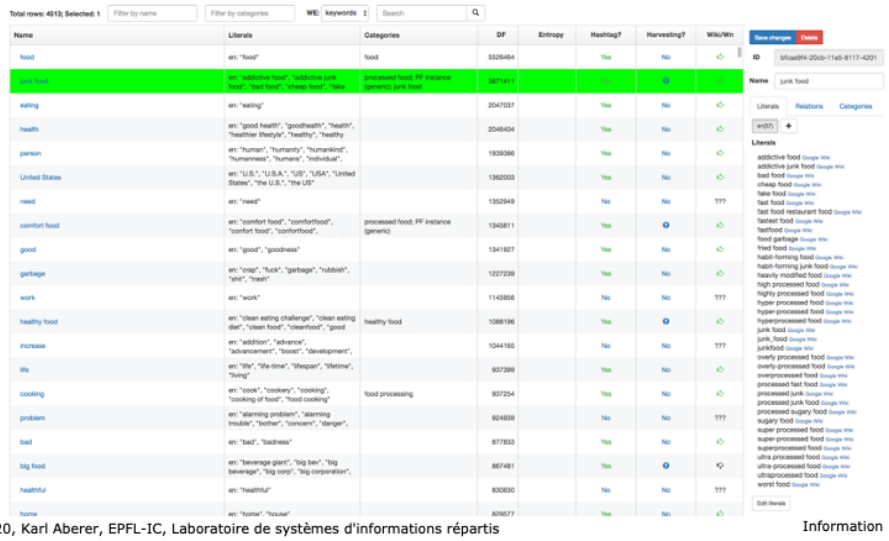
- Use of many structural, syntactic features of the documents
- Use of external resources, such as Wikipedia, Wordnet

The basic approach to key phrase extraction is based on principles well known from information retrieval. A first decision is what should be possible key phrases. As in IR stopwords are excluded and key phrase candidates could be all n-grams from the remaining text (where n is typically in the range of 2 to 5). Furthermore part-of-speech tags could 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, basic tf-idf ranking is a possible approach. As in IR a candidate phrase is considered as relevant for the document if it is at the same time frequent and distinctive. Apart from that, many heuristics have been developed to refine this approach, taking into account additional features of the document. For example, a phrase in the title or a header could be considered as more relevant. Or, external knowledge bases could be used to check whether a phrase corresponds to a commonly known concept.

Use of Keyphrase Extraction

Creation of domain-specific thesaurus and taxonomy

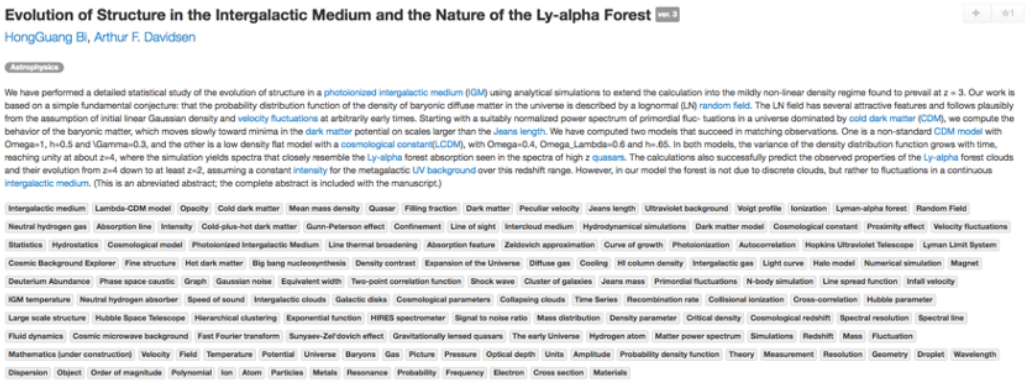


Key phrase extraction can be used as an initial step to create a domain-specific thesaurus or taxonomy. In the example, we see a thesaurus that has been constructed for the food domain, based on a key phrase extraction. For the phrase “junk food” a large number of synonyms and near-synonyms have been 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, among others, to retrieve documents that refer to this concept with higher recall, than would be possible with a simple keyword search just using the phrases “junk food”, and possibly some alternative phrases found in an ad-hoc way.

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 the result of key phrase extraction for scientific documents in physics. A large number of 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.

2. NAMED ENTITY RECOGNITION

Named Entity Recognition (NER)

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

organization organization

EPFL is one of the two **Swiss Federal Institutes of Technology**. With the status of a national school since 1969, the young engine[georegion] has grown in many dimensions, to the extent of becoming [city]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 stepping stone to extract more complex pieces of knowledge, in particular statements, 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 in particular 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 a number of commercial 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**

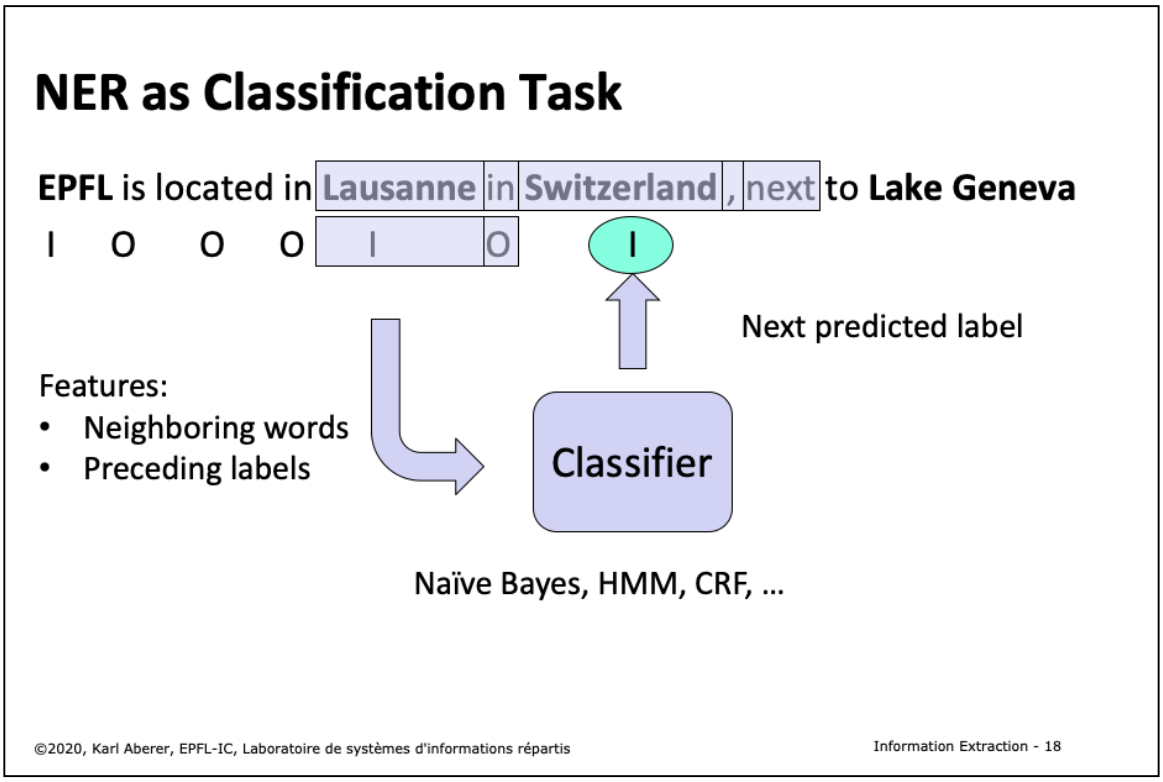
I O O O I O I O O O I I

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 a 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 entity, 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, in particular whether a word is the beginning or end of entity name, could also be done. Note that in this context also punctuation marks are considered as words, as they may carry important information on the presence of an entity.



Given that NER can be understood as a classification problem, we have to answer two questions, namely which are the input features for the classifier and 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 a number of derived features can be produced. One additional type of feature that is specifically used in NER is the labels that have been produced by the classifier for the words preceding the word to be classified. Thus, the classifier classifies words while reading the words in the sequence they appear in the document. Thus, even though in principle any classifier could be applied, e.g. Naïve Bayes, which is used frequently in practice, 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

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 allows 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, thus learning this could 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 a important features, as usually names start in 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

Generative Probabilistic Model

Sequence of words (known): $W = (w_1, w_2, w_3, \ldots, w_n)$

Sequence of labels (unknown): $E = (e_1, e_2, e_3, \ldots, e_n)$

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 now will see a classifier that specifically takes advantage of the sequential nature of learning entities in natural language text. The approach is strongly related to probabilistic information retrieval and based on a generative probabilistic model for text.

The basic model assumes that there exists a (unknown) probability distribution $P(E, W)$ that connects sequences of words with the corresponding sequences of labels. The task of classification is then to identify for a given sequence of words, the most probable sequence of labels given the sequence of words. 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 of how the different labels interact with each other, and P(W|E) is a model of how words interact with labels.

Approximation

Label transition probabilities (bigram model)

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

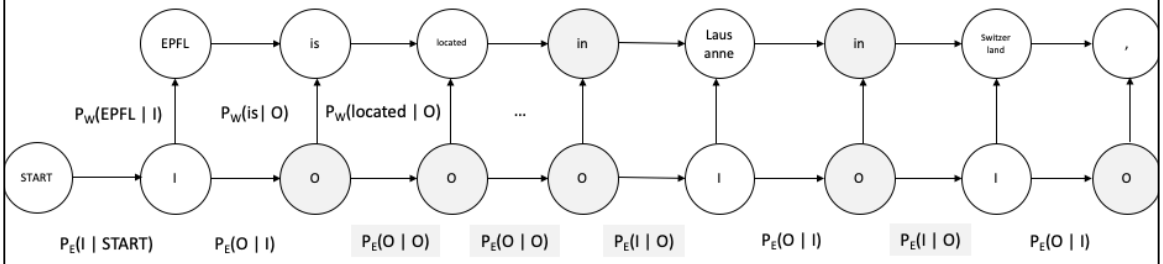
Word emission probabilities

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

As we will not be able to estimate the complete probability distribution functions, we approximate them by making (sever) 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 assumed in probabilistic information retrieval). For words we assume that their probability depends only on the label that it received. Thus the two probability functions decompose into products of much simpler functions that we can estimate.

Hidden Markov Model (HMM)

Graphical representation of the approximate probabilistic model



Maximum Likelihood Estimation

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

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, we have here the case of a hidden Markov model since the labels E are unknown and their probabilities need to be estimated from the known words.

The approach for HMM can be applied to any sequence labelling task. In particular it can be used to learn part-of-speech tags and the types of the entities.

Learning the Model

Maximum Likelihood Estimation requires only counting

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

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

For estimating the basic probabilities $P_E(e_i|e_{i-1})$ and $P_W(w_i|e_i)$ we use maximum likelihood estimation as in probabilistic information retrieval. For example, for estimating a 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 was I respectively O with the total number of occurrences.

As in probabilistic information retrieval we have the issue of sparsity of words in the training set. As a result it might occur that for a specific label no words appear as examples in the training data, whereas it is not excluded that such a word might be related to the label in general. With smoothing this is taken into account, where the smoothing parameter depends on the length of the word sequence.

For the labels no smoothing is required, as the number of labels is very small (in the case of untyped entity recognition it is 2) and thus all label combinations are extremely likely to occur.

Using the Model

For a given sequence of words W find the most probable model 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

For using the HMM Model the problem is of simply finding the sequence of labels that is the most probably one produced by the HMM for a giving sequence of words. In principle, this can be performed by brute-force search, but which does not scale for longer text due to a combinatorial growth in cost.

Observation

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

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

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

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

A simple observation is that the choice of the last label in a sequence that produces the largest probability can be made independently of the choices of the preceding labels that produce a maximal probability. Based on this observations the computation of the sequence with the maximal probability can be dramatically simplified.

Viterbi Algorithm

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

Then

$$\pi(k, v) = \max_u \pi(k-1, u)\ P_E(e_k|u)\ P_W(w_k|v)$$

$$\pi(0, *) = 1$$

This is a dynamic programming algorithm
→ Viterbi algorithm

It is just necessary to keep the information which is the sequence that produces the maximum value up to the last label, and then compute the label that maximizes this probability in the last step. In the case where the HMM model considers only the value of the previous label for predicting the next label this results in the aboce algorithm. In its generalized form, where labels can depend on multiple other labels (resp. random variables) the algorithm becomes a dynamic programming algorithm and it is called the Viterbi algorithm according to its inventor.

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

3. INFORMATION EXTRACTION

Information Extraction (IE)

Task: Extract statements from text
→ creation of knowledge graphs

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)**.

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.

Taking the analysis of documents one step further, we now consider the extraction of statements from natural language text, as is illustrated in the example. Statements connect entities through relationships, for example, the first statement expresses that EPFL is part of a larger organization, the Swiss Federal Institutes of Technology. Thus the notion of statements we use here corresponds exactly to the notion of statements we introduced earlier with RDF.

Sample Statements

EPFL is one of the two **Swiss Federal Institutes of Technology**

EPFL - IS-A - Swiss Federal Institute of Technology

its sister institution in **Zurich**, **ETHZ**

EPFL - RELATED-TO - ETHZ

EPF domain , which is directly dependent on the **Federal Department of Economic Affairs**

EPF Domain - DEPENDS-ON - FDEA

EPFL is located in **Lausanne**

EPFL – LOCATED-IN - Lausanne

Lake Geneva and at the foot of the **Alps**

EPFL – LOCATED-IN - Alps ? Lake Geneva – LOCATED-IN – Alps?

Looking more closely at some of the statements we can extract from the text, we make the following observations. First, statements are always anchored in two entities, thus the link two entities by a relationship. Second, the relationships can carry different meanings, which in natural language are typically expressed in verbs. Third, the extraction of statements can be ambiguous. In the last example, when looking at the original text, the implied meaning is that EPFL is close to the alps. When looking only at the local context of ”Lake Geneva” and “Alps”, one might make also the (incorrect) inference that the statement is about Lake Geneva located in the alps. Accidentally this is not a wrong statement, but not the one that is intended in the text. So statement extraction can be a tricky task due to the ambiguity and complexity of human language.

Typed Statements

EPFL – PART-OF – Swiss Federal Institute of Technology

Type: ORG – PART-OF – ORG

EPFL – RELATED-TO – ETHZ

Type: ORG – RELATED-TO – ORG

EPF Domain – PART-OF – FDEA

Type: ORG – PART-OF – ORG

EPFL – LOCATED-IN - Lausanne

Type: ORG – LOCATED-IN – LOC

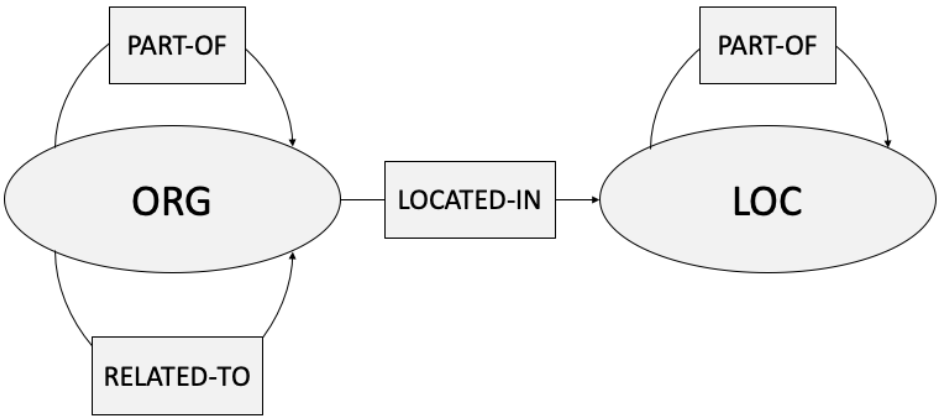
EPFL – LOCATED-IN - Alps ? Lake Geneva – LOCATED-IN – Alps?

Type: ORG – LOCATED-IN – LOC

Type: LOC – LOCATED-IN – LOC

We can from the individual statements generalize to the statements with the type of entities, resulting in statement types. This implies that we may assume the existence of a schema for statements (in RDF an RDF schema) that specifies which types of entities can be connected by which types of relationships. Not every type of entity can be related in a meaningful way to another type of entity. For example, it would not make sense to have a statement expressing that a location is part of a person. Having these additional constraints, will be helpful to more precisely detect statements.

Statement Schema



This is an example of a possible schema that constrains the type of statements to be considered.

Approaches to Information Extraction

1. Hand-written patterns
2. Supervised machine learning
3. Bootstrapping
4. Distant supervision
5. Matrix Factorization

1. Hand-Written Patterns

Early approach from **Hearst (1992)**

- "Agar is a substance prepared from a mixture of **red algae, such as Gelidium**, for laboratory or industrial use"

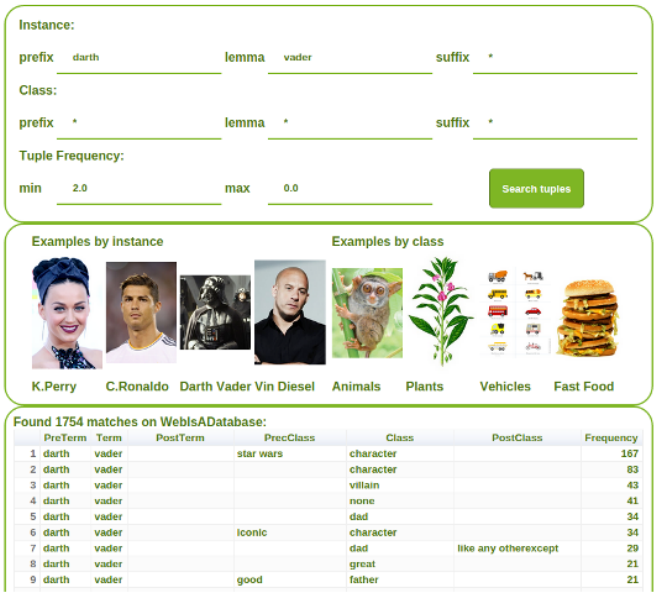
Patterns to detect IS-A relationships:

```
"Y such as X ((, X)* (, and|or) X)"
"such Y as X"
"X or other Y"
"X and other Y"
"Y including X"
"Y, especially X"
```


A first approach to statement extraction is based on the observation that in natural language often a relationship is expressed in a regular fashion. Very early on, this has been used to extract specific relationships, such as ISA, by using regular expression patterns. Hearst was one of the first attempts taking this approach, and the method is still being used till today, potentially with some expanded set of patterns.

Web isa Database

Large scale extraction of IS-A relationships from web documents



	PreTerm	Term	PostTerm	PrecClass	Class	PostClass	Frequency
1	darth	vader		star wars	character		167
2	darth	vader			character		83
3	darth	vader			villain		43
4	darth	vader			none		41
5	darth	vader			dad		34
6	darth	vader		iconic	character		34
7	darth	vader			dad	like any otherexcept	29
8	darth	vader			great		21
9	darth	vader		good	father		21

http://webisadb.webdatacommons.org/webisadb/

Web is a DB is an example a large-scale effort to extract IS-A relationships from Web data. The patterns are more complex than the ones introduced by Hearst.

More General Hand-Written Patterns

Idea: relations often hold between specific entities

- located-in (ORGANIZATION, LOCATION)
- founded (PERSON, ORGANIZATION)
- cures (DRUG, DISEASE)

First, perform Named Entities Recognition

Use typed pattern: ORG is located in LOC, LOC

EPFL is located in **Lausanne**, **Switzerland**

The idea of Hearst that has been successfully applied for detecting ISA relationships, can be extended to patterns of a more general type, for extracting statements for other types of relationships. I this approach, one can exploit the fact that certain relationships can only hold among certain types of entities. Thus in a first step a named entity recognition is performed, and subsequently the patterns are searched for which the matching types of entities can be found.

Summary Hand-Written Patterns

Advantages

- Rules tend to be high-precision
- Can be tailored to specific domains

Disadvantages

- Human patterns are often low-recall
- A lot of effort to think of all possible patterns

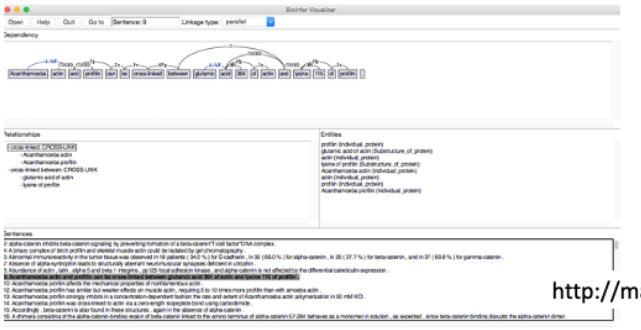
Hand-written patterns is in general a very reliable method for statement extraction. However, it suffers from low recall and it is very difficult to conceive all possible patterns by a human (expert). Therefore more automated methods are of interest, that we will discuss in the following.

2. Supervised Learning for IE

Approach: train a classifier on labeled data

Creating a training set

- Choose relevant named entities and their relations
- Hand-label relations among entities (positive examples)



http://mars.cs.utu.fi/BioInfer/

The supervised learning approach for information extraction requires first a training set. Producing such a training set is labor-intensive and a major task. For example, for the domain of life sciences efforts have been undertaken. In the example shown, more than 2000 sentences have been manually annotated.

Classifiers for Information Extraction

Two-step approach

- A **filtering classifier** (e.g., Naïve Bayes), to detect whether a relation exists among the entities
- A **relation-specific classifier** detecting the relation label

Training the classifiers

- Extract named entities in the document corpus using NER
- Detect pairs of entities, e.g., in the same sentence
- Use unlabeled entity pairs as negative examples

Once a manually annotated document collection is available, classifiers for information extraction can be trained. One approach is to train two types of classifiers, one that simply detects whether a relationship exists among two entities, and a second that detects the type of the relationship. The use of a filtering classifier can speed up the classification task and allows the use of distinct feature-sets for the two tasks.

For training the classifiers, one first extracts all entity pairs using NER that occur in the same context, for example, in the same sentence. If the pairs are not annotated they are taken as negative examples. Then the classifier is being trained using features extract from the context of the occurrences of the entities.

Features Used in IE

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

Features for mention (M1, M2) = (Lausanne, Lake Geneva):

BOW and bigrams in the sentence:	is, located, in, located in, Lausanne, in, next to …
BOW and bigrams in between the mentions:	in, Switzerland, next, to, next to
Headwords* of mentions, their concatenation:	Lausanne, Geneva, Lausanne-Geneva
Words in positions:	M1-1: in, M1+1: in, M2-1: to
Types of entities	LOC, LOC
Stemmed version of the words	
Syntactic features	
…	

*headword = entry in a dictionary

The features being used for information extraction are typically different from the ones used for NER. When we have identified two occurrences of named entities in a sentence, also called two mentions (M1, M2), then we can identify the following features related to the two mentions:

- The bag of words and bigrams found within the whole sentence
- Distinct from that the BOW and bigrams in between the mentions
- The headwords (these are words that are typically found in a standard dictionary)
- Words in specific position with respect to the mentions
- Stemmed versions of all the words above
- The type of entities used
- Syntactic features, extracted with part-of-speech analysis

Syntactic Features

Parse Tree

```
(S (NP EPFL)
        (VP is
                (VP located
                        (PP in
                                (NP Lausanne))
                        (PP in (NP Switzerland ,
                                (PP next to
                                        (NP Lake Geneva)))))))
```

Features:

Sequence between entities: PP NP PP NP

The syntactic features, or part-of-speech tags can be exploited in various ways. In the simplest case only the sequence of POS tags in between the mentions is used. More complex features can be constructed as well, e.g. the navigation path between the mentions in the parse tree. For trying out POS tagging you can try: http://www.link.cs.cmu.edu/cgi-bin/link/construct-page-4.cgi#submit

Which is true?

A. Hand-written patterns are in general more precise than classifiers
B. Hand-written patterns cannot exploit syntactic features
C. Supervised classifiers do not require any human input
D. Supervised classifiers can only detect typed statements

3. Bootstrapping

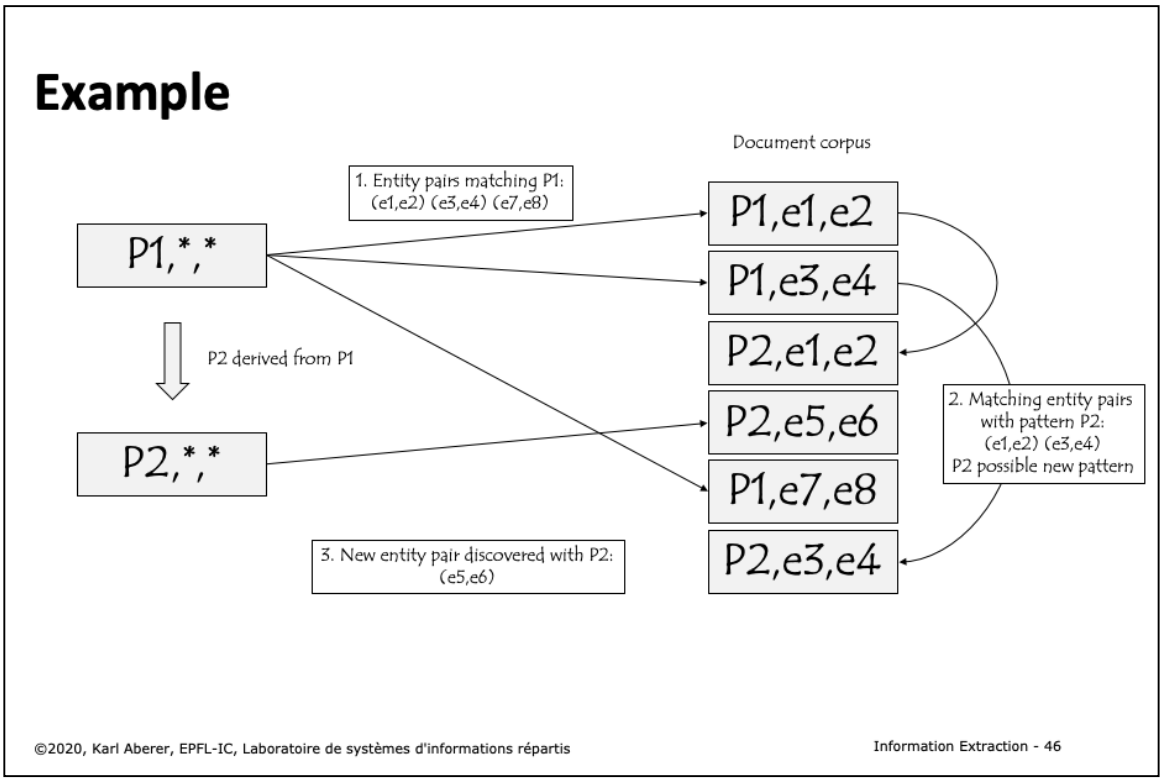
No training data, but a few high-precision patterns

Approach:

- Find entity pairs that match the pattern
- Find sentences containing those entity pairs
- Generalize the entities in those sentences
- Generate new patterns

One of the big problems with supervised approaches to information extraction is the scarcity of training data. One way to approach this problem is called bootstrapping. The basic idea is that one may have a few high-precision patterns, such as the Hearst patterns, or simple a set of example statements that are manually extracted, and one tries to generalize these patterns by analyzing a large document corpus.

The approach is to first find entity pairs using the high precision patterns. Then using those entity pairs sentences containing the same entity pairs are searched. The assumption is that with large likelihood these sentences express the same type of relationship, just in a different syntactic representation. Thus such sentences can be considered as templates for expressing the relationship. By generalizing the occurrences of the entities in such a sentence, now patterns for identifying the relationship are produced.



This graphics illustrates the different steps of how new patterns are detected and applied to find new relationships.

Example

Pattern: **LOC** is located in **LOC**

- Mumbai **is located in** India
- Adelaide **is located in** Southern Australia
- Sriharikota **is located in** Nellore

Search for entity pairs (Mumbai, India)

- **Mumbai** is **India's** top destination
- **Mumbai** hotels, **India**

New patterns

- **LOC** is **LOC**'s top destination
- **LOC** hotels, **LOC**

This example illustrates the approach. We start with a simple, but precise pattern, LOC is located in LOC, with which we find occurrences of entity pairs. Then we search for those entity pairs, and find other sentences mentioning them. These are then generalized to patterns by replacing the entity occurrences by their types.

Problem: Semantic Drift

Example: The pattern

LOC hotels, **LOC**

matches also

… Geneva hotels, Lausanne hotels …

➔ Geneva is located in Lausanne?

Of course, this approach risks to infer wrong rules for many reasons. For example, by matching the patterns without considering the context wrong inferences as the one shown in this example could be made.

Confidence

Assume we have a confirmed set of pairs of mentions M

- A new pattern should also match many of those

$Hits_p$ = number of pairs in M that a new pattern matches

$Finds_p$ = total number of pairs that a new pattern matches

Confidence that a new patterns finds many relevant mentions

$$Conf(p) = \frac{Hits_p}{Finds_p} \log(Finds_p)$$

In order to contain the problem of inferring too many ”wrong” patterns, a confidence metric can be used. With this metrics the confidence in a new template increases, whenever it retrieves more entity pairs that are already confirmed to be correct. However, this is balanced by measuring of how many entity pairs the new template matches in total. If in proportion it matches too many pairs, without creating confirmed entity pairs, the confidence in the template is lowered.

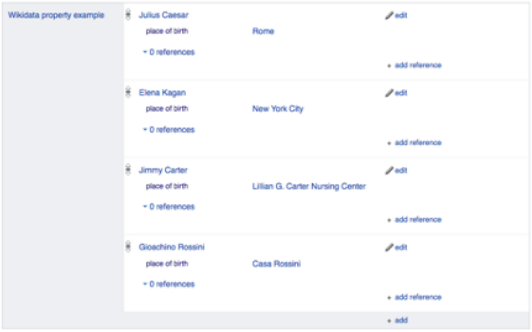
4. No Training Data? Distant Supervision

Idea: use existing knowledge bases to collect training data for building a classifier

- Combines advantages of bootstrapping with supervised learning

Example: learning PLACE-OF-BIRTH

For example, WikiData has many positive examples!



Another idea to deal with the problem of lacking training data is based on the observation that large databases of known statements (also called facts) do exist. One example is WikiData. Thus, instead of producing confirmed statements from a training corpus by using high precision patterns, such statements are drawn from an existing knowledge base. In this way, large document collections can be searched for sentences that potentially express those facts, and these sentences can be used to train a classifier for information extraction.

Linking Text to Knowledge Bases

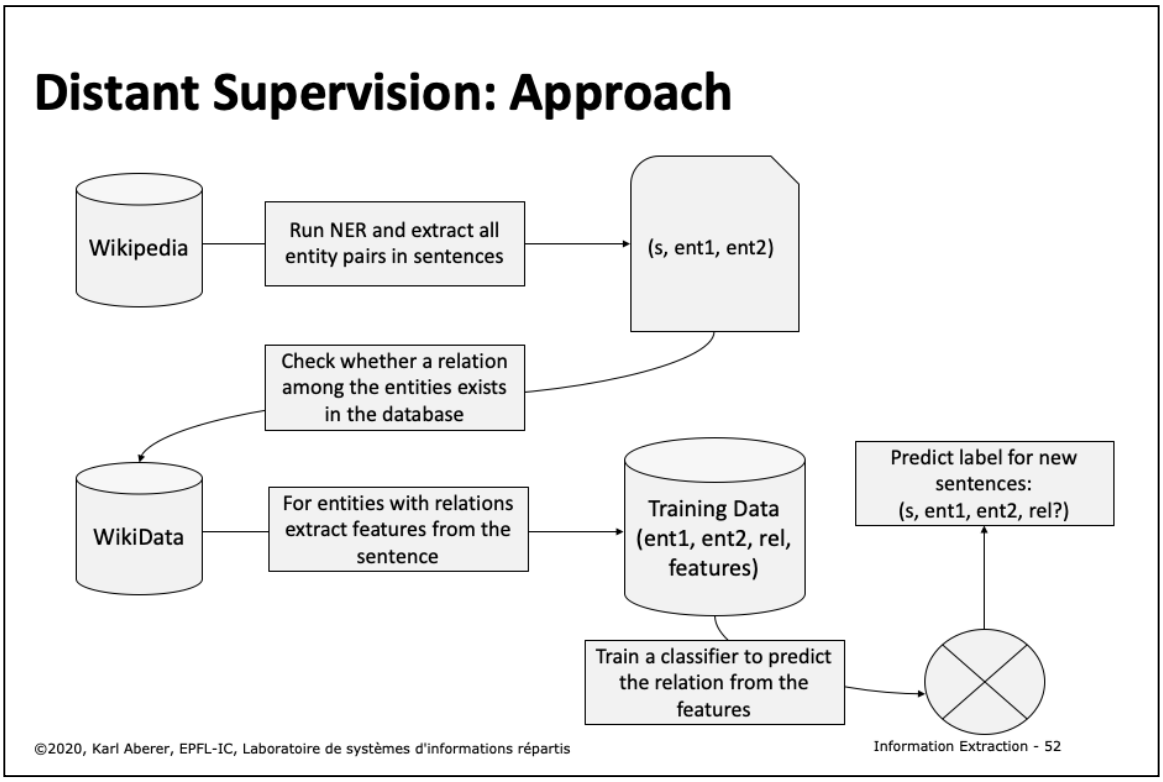
Using Entity extraction, entity mentions in text can be linked to corresponding mentions in a knowledge base

John was born in Liverpool, to Julia and Alfred Lennon

Entities		Text Features (e.g. pattern)			Relation from knowledge base	
Entity 1	Entity 2	PER was born in LOC	PER was born to PER	PER and PER	Birthplace(X,Y)	Married(X,Y)
John Lennon	Liverpool	x			?	
John Lennon	Julia Lennon		x			
John Lennon	Julia Lennon		x			
Julia Lennon	Julia Lennon			x		?
Barack Obama	Hawaii	x			x	
Barack Obama	Michelle Obama			x		x

Barack Obama was born in Hawaii. Barack and Michelle Obama …

In this example we illustrate of how entity pairs identified in text can be linked to the same entity pairs as they are found in a knowledge base. This enables to reason about the meaning of certain syntactic patterns (e.g. ”was born in”) and infer what their formal meaning could be (e.g. birthplace).



Here we illustrate the process of the distant supervision method. First, we start by running NER over a large document collection, such as WikiPedia. This produces a set of sentences s that contain entity pairs (e1,e2). Having those entity pairs, next we check in the existing knowledge base, in this example WikiData, whether relations among the entity pairs exist. For those entity pairs, for which a relation is confirmed in the knowledge base, we have identified an instance for the training data, and we extract all typical features used for information extraction from the sentence in which the entity pair has been found. This produces a training set for training a classifier to predict the relation. Finally this classifier can be used to predict relations for unseen documents, by extracting occurrences of entity pairs and feeding the classifier with the features of the sentence in which they are contained.

Features for Distant Supervision

Use conjunctions of standard IE features as sentence features

- Match only if all individual features match
- High precision, but low recall features!
- Feasible, since training set is large

Complex features resemble to templates used in rule-based approaches

In distant supervision the features are constructed in a different way than in standard information extraction using supervised learning, due to the fact that the number of training examples is in generally much higher. Instead of producing a feature vector from combining all individual features, each feature combination is considered as a separate feature, which results in a much larger feature space. As a result, those more complex features are much more precise, but have low recall. This is, however, compensated by the fact that the training set is much larger.

Example

Complex Features for **"EPFL** is located in **Lausanne** in Switzerland":

F0: M1=ORG and M2=LOC and betweenwords={is, located, in} and afterwords={} ← Window size 0
and betweenPOS = {VP, VP, PP}

F1: M1=ORG and M2=LOC and betweenwords={is, located, in} and afterwords={in} ← Window size 1
and betweenPOS = {VP, VP, PP}

→ F0 matches **"ETHZ** is located in **Zürich**, Switzerland", F1 not

Individual features:

F1: M1=ORG, F2: M2=LOC, F3: betweenwords={is, located, in} , F4: afterwords0={},
F5: afterwords1={in}, F6: betweenPOS = {VP, VP, PP}

→ only F5 is different

The example illustrates this point. We construct two complex features by computing the conjunction of several simple features, once considering a window of size 0 around the phrase containing the entities (resulting in feature F0) and once considering a window of size 1 around the phrase containing the entities (resulting in feature F1). As a result, the complex feature F1 does not match at all the sentence "**ETHZ** is located in **Zürich**, Switzerland".

If used a feature vector derived from the individual features, the two feature vectors derived for the two documents would be very similar for these feature set, since most of the features are identical. Only feature F5 would be different.

Which is true?

A. Distant supervision requires rules for bootstrapping
B. Classifiers produced with distant supervision are more precise than rules
C. Distant supervision can help to detect rules

5. Matrix Factorization

Using the same data as for distant supervision

- Entity pairs from text linked to relations from knowledge bases

Instead learning a classifier, create low-dimensional representations for entity pairs and relations

Use those representations to

- Link text patterns to relation types and identify similar text patterns
- Extract relations from text

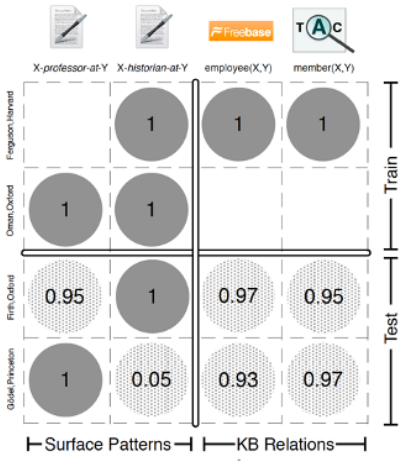
Distant supervision aims at generating classifiers for relations that are based on syntactic features. With the same information used for distant supervision we could also try to understand the intrinsic nature of relationships by mapping them to a low-dimensional representation, as we did earlier for words with word embeddings. This is the idea underlying to apply matrix factorization ot the entity-pairs / relation matrix.

Matrix Representation

Create a matrix with

- Entity pairs as rows
- Relation types as columns
  - Relations from text patterns
  - Relations from knowledge base

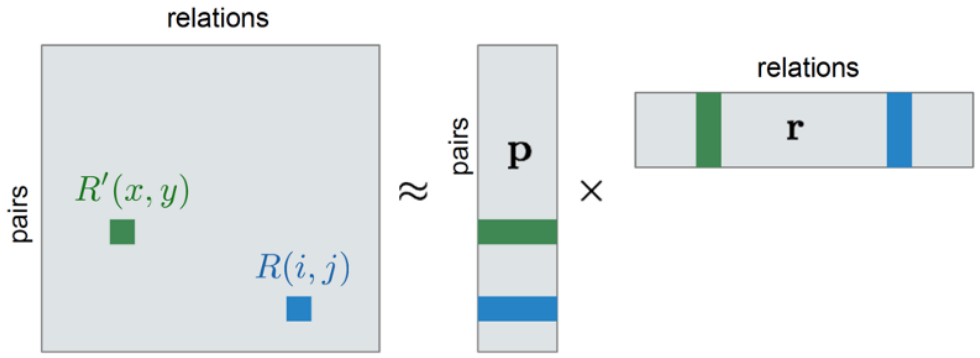
The entity-pair/relation matrix is a sparse matrix (like in recommende systems)



By using matrix factorization the hope is to align similar text patterns and corresponding relationships in a latent space. This can help both in identifying text patterns that correspond to relationships and to extract those relationships from text.

The entity-pair/relation matrix is a sparse matrix is a sparse matrix. Thus situation is comparable to the one we had in recommender systems. Algebraix factorization methods would not work. On the other hand using matrix factorization based on SGD might help to “guess” new relationships (as in recommender systems it helps to guess unseen ratings). This idea we will alter exploit also for the problem of link prediction.

Matrix Factorization



$R(i,j)$ positive examples of facts

$R'(x,y)$ negative examples of facts

The matrix factorization will represent both entity pairs and relations as low-dimensional vectors in a latent space. The entries in the matrix will be the corresponding scalar products. The entries in the matrix are high probabilities if a relationship holds, and low probabilities if this is not the case.

From distant supervision and text analysis we have positive examples. For learning we also need negative examples.

Bayesian Personalized Ranking

Idea: give observed true facts higher ranking then unobserved (true or false) facts

Approach: create ranked pairs f^+ and f^-

Objective Function

$$\sum_{f^+,f^-} \log \sigma\left(\theta_{f^+} - \theta_{f^-}\right)$$

where

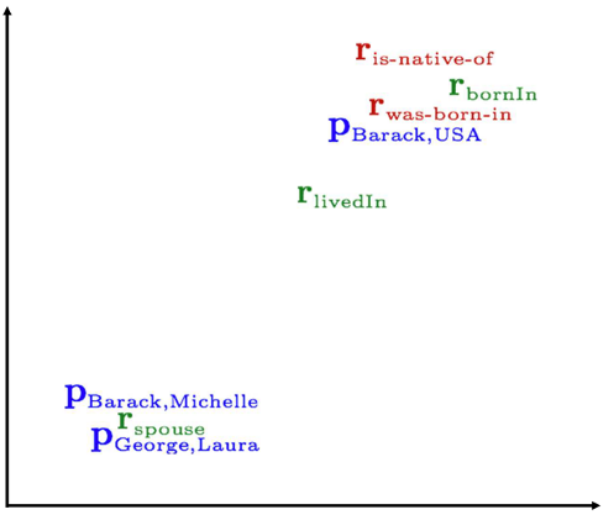
$$\theta_f = \boldsymbol{p} \cdot \boldsymbol{r}$$

Maximize with Stochastic Gradient Descent

For choosing negative examples of facts we encounter the problem that we can choose any fact that has not been observed in the data as such a negative example, but that we cannot be sure that the chosen example really is a negative example. It could simply be a relationship that holds, but has not been registered anywhere. This problem is similar to the one encountered in recommender systems where the absence of a rating does not imply a negative rating.

To better adjust to this situation an alternative method for matrix factorization, called Baysian Personalized Ranking, is used. It is based on an alternative loss function is constructed that attempts to maintain the relative ranking among positive and negative examples.

Relation Embeddings



The matrix factorization obtained this way allows to map entity pairs and relationships in the same low-dimensional space. As illustrated in this example this should cluster together those that coorespond to similar relationships. This allows to infer new relationships for existing entity pairs, as well as syntactic forms of relationships.

Exploiting Relation Embedding Similarity

Similar in embedding space

Similar in embedding space

Entities		Relationship pattern	
Entity 1	Entity 2	COM owns part in COM	COM buys stake in COM
Renault	Nissan	x	x
BMW	Rover		x
Volkswagen	Porsche		
Ford	Toyota		

Possible inferences:

- BMW owns part in Rover (similarity of relationship)
- Volkswagen owns part in Porsche (similarity of entity pair)

This example illustrates of how the method could be used to extract relationships for previously unknown syntactic patterns.

Summary

Information extraction

- Populating knowledge bases and fact databases
- Taxonomy induction

Pattern-based approaches

- High precision, low recall, work intensive

Supervised learning

- Low precision, high recall, work intensive

Hybrid methods: bootstrapping, distant supervision, matric factorization

No statement schema known: open information extraction

We have discussed the different variants of information extraction methods typically used, with their advantages and drawbacks. Another problem of information extraction is called open information extraction. It is typically used for extracting statements from large collections of documents, such as the Web, where no predefined schema of relations and entity types is known. Such methods have first of all to be able to identify that a statement is present. Such methods rely on significant syntactic preprocessing of the text, and use verbs as anchor points to detect statements.

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*.