Proceedings of the Federated Conference on Computer Science and Information Systems pp. 471–478
DOI: 10.15439/2018F168
ISSN 2300-5963 ACSIS, Vol. 15

What was the Question? A Systematization of Information Retrieval and NLP Problems

Jens Dörpinghaus*, Johannes Darms† and Marc Jacobs‡
Fraunhofer Institute for Algorithms and Scientific Computing,
Schloss Birlinghoven, Sankt Augustin, Germany
Email: *jens.doerpinghaus@scai.fraunhofer.de, †johannes.darms@scai.fraunhofer.de, ‡marc.jacobs@scai.fraunhofer.de

***Abstract*—In this paper we suggest a novel systematization of Information Retrieval and Natural Language Processing problems. Using this rather general description of problems we are able to discuss and proof the equivalence of some problems. We provide reformulations of well-known problems like Named Entity Recognition using our novel description and discuss further research and the expected outcome. We will discuss the relation of two problems, cluster labeling and search query finding. With these results we are able to provide a novel optimization approach to both problems. This novel systematization approach provides a yet unknown view generating new classes of problems in NLP. It brings application and algorithmic approaches together and offers a better description with concepts of theoretical computer science.**

I. Introduction

A LOT of research in the last decades focused on the computational perspective of information retrieval and improving clusterings, partitions, search queries and document decomposing with and without feedback. Several authors like Manning et al. [1] or Clarc et al. [2] give an overview about the algorithmic part of computational linguistics and NLP. Applied researchers try to answer questions similar to "What was the question?", "What is the best description of this set of documents?", "How can we compare this and that clustering?". We realized that there are several names for the same or at least similar problems and approaches. For example Hagen et al. [3] tried to find search queries for a given set of documents and used it as a cluster labeling approach. This seems somehow obvious, as well as some other equivalent problems might also sound obvious. But nevertheless, we think a formal description and proof is necessary.

During our literature search and evaluation of several algorithms for query optimization and clustering, we realized that a formal Schema would ease the task. For finding and grouping This we propose such a schema within this work. We claim every NLP problem can be described using a five-tuple. To proof our theory a discussion on several problems and the proof of equivalent problems will follow. This novel systematic approach has a different perspective focusing on the computational view on this research area. We hope this early research will lead to a valuable discussion and more research on the theoretical and algorithmic fundamentals of natural language processing.

In table I we list some prominent NLP and IR probleme in our proposed five tuple with a corresponding description. The details of the tuple are introduced and discussed in the following chapters. However the connection between the problems can already be seen within this table.

TABLE I: Example formulations of information extraction problems as five-tuple. The first element describes the domain set, the second the domain subset of interest. The third element is a description function f. We either note this function or the image set of this function. The last entries are a feasible similarity or error measure and a reference standard. These problems are introduced in sections III and IV.

Problem Formulation	Problem Description
$\mathbb{D}\|R\|\mathbf{X}\|err\|R$	Generating of optimal Search Queries
$\mathbb{D}\|R\|\mathbf{X}\|err\|R$	Generating of optimal Cluster Labels
$\mathbb{S}\|\emptyset\|\boldsymbol{f}\|e\|\{\mathbb{S}\times[0,1]\}$	Named Entity Recognition
$\mathbb{D}\|R\|\mathbf{L}\|sim\|\emptyset$	Text summarization
$\mathbb{D}\|R\|\mathbf{K}\|sim\|\emptyset$	Keyword identification
$\mathbb{D}\|R\|\mathbf{C}\|sim\|\emptyset$	Document Clustering in $C=\{1,...,n\}$ cluster.
$\mathbb{D}\|R\|\mathbb{D}^{\mathbb{D}}\|sim\|\emptyset$	Relation Extraction
$\mathbb{D}\|R\|\mathbb{D}\|sim\|R$	Document Subset Finding Problem
$\mathbb{D}\|R\|\mathbf{G}\|sim\|\emptyset$	Parse tree

II. Notation

We want to introduce our problem description approach using a five-tuple. Therefore we define a domainset $\mathbb{D}$ and subset $R\subseteq\mathbb{D}$, a description set $\mathbb{X}$ and a description function $f:\mathbb{D}\rightarrow\mathbb{X}$, an evaluation function $e:\mathbb{E}\rightarrow[0,1]$ and a reference standard E. Hence NLP problems can be given as:

$$p=\mathbb{D}|R|f:\mathbb{D}\rightarrow\mathbb{X}|e:\mathbb{E}\rightarrow[0,1]|E \tag{1}$$

At first we have to introduce and describe the notation and sets. Some examples and applications will be provided within the next sections. For an illustration of sets and functions we refer to figure 1.

A. Domain Set

Let $\mathbb{D}$ be a finite *domain set* containing all instances of a Probleme, e.g documents, text data, speech or any other semantic content. A definition is $\mathbb{D}=\{d_1,...,d_n\}$ where d_i is a vector of documents or semantic data. We may see a textual

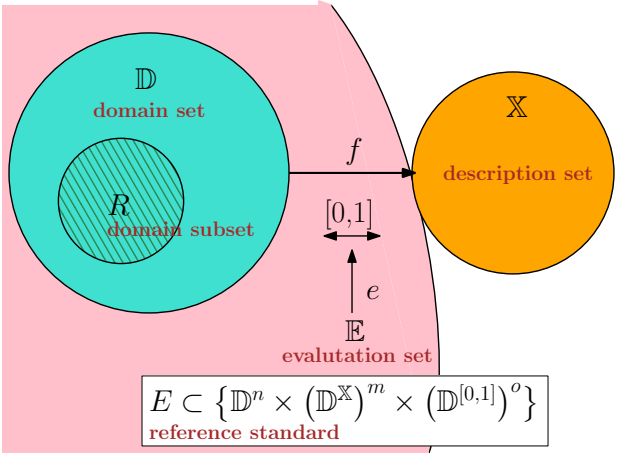


Fig. 1: Illustration of sets in equation 1. If not feasible, these sets or functions can be set to $\emptyset$ or id. The cut between $\mathbb{X}$ and the other sets is not necessarily empty.

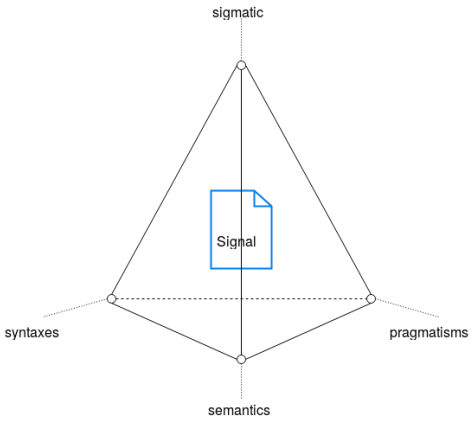


Fig. 2: Illustration of an SDA. All four properties, (a) Sigmatics (identification) (b) Pragmatics (what is being represented) (c) Semantics (what is being represented) and (d) Syntax (how is the signal constructed), are wrapped around the digital asset to provide more meta-data. All five elements for the semantic digital asset which. Multiple SDAs connect if they share at least one of these elements.

document d as a vector containing n meta data as well as full text etc., describing a document as follows

$$d_i = \begin{pmatrix} d_i^1 \\ d_i^2 \\ d_i^3 \\ \vdots \\ d_i^{n_i} \end{pmatrix} = \begin{pmatrix} \text{title} \\ \text{authors} \\ \text{fulltext} \\ \vdots \\ \text{NE} \end{pmatrix}$$

Here $\mathbb{D} = L^{\mathfrak{n}}$ with $\mathfrak{n} = \max_i n_i$ is the vector space of $\dim(\mathbb{D}) = \mathfrak{n}$ data fields of a natural language L. We may also store binary data within this vector. For a better generalization we can set $\mathbb{D} = \mathcal{P}(\mathbb{S})$ as the vector space of semantic digital assets. Here $\mathcal{P}$ denotes the power set. In this case a document d is a list $d = \{s_1, ..., s_n\}$ of semantic digital assets (SDA) $s_i \in \mathbb{S}$. These SDAs are an optimal carrier for meta-data or annotations.

A semantic digital asset can be defined as "an asset that exists only as a numeric encoding expressed in binary form" [4]. This definition includes text, images, sound files, tables, and so on. In a nutshell we can conclude that "digital assets include any electronically stored information" [5]. In addition some meta data is included. Thus we can describe a SDA as a variation of the information tetrahedron introduced in [6] where four semiotic properties are wrapped around each signal. These semiotic properties are (a) Sigmatics (b) Pragmatics (c) Semantics and (d) Syntax, see figure 2. As described by Hodapp et al. in [7] SDAs are highly flexible and can be easily connected. The hierarchy is connected into annotations. Applications can be found in [7] and [8].

Since it is of crucial importance, we will discuss the hierarchical connection in a nutshell, but refer to [7] for further information. Let a and b be defined with Sigmatics *sentence:S153:1322066041* and *sentence:S153:1322066041* – the first one with the Pragmatics *sentence* and the second one *Mus_musculus*. Here, the first SDA contains the sentence in natural language, for example "Spontaneous antepartal RhD alloimmunization" whereas the signal of the second one contains the named entity information *{"begin": 24,"end": 27,"attr": "original","ref": "MM137098:Rhd"}*. Thus it is not really necessary to store explicit relations between SDAs, since they are implicitly given in the structure of SDAs.

B. Domain Subset

To find a proper problem description, we can either focus on the complete domain set $\mathbb{D}$ or a subset $R \subseteq \mathbb{D}$ This can either be a manually created subset or created by semi-automatic tools. One can imagine the result of a search query.

In addition, we allow a ranking of elements in R in the intervall $[0, 1]$ of real numbers. Then $R = R' \times [0, 1]$ with $R' \subseteq \mathbb{D}$.

C. Description Function

If necessary, we may also add a *description function* f for documents or subsets of $\mathbb{D}$. Given a description set $\mathbb{X}$, this function can have several forms, in general denoted by $f : \mathbb{D} \to \mathbb{X}$. In our short notation we can either note the function f or the set $\mathbb{X}$ if we need to focus on this set. If both information are needed, we can write $f, \mathbb{X}$.

If we want to map elements in R to a meta data subset of a document $d \in \mathbb{D}$ like publishers, authors etc. f has the form $f : \mathbb{D} \to D$ with $\dim(D) \leq \dim(\mathbb{D})$ and $f(d_i) = f_i^j$. This may also be a combination of vector entries.

A description function may also return several discrete values, for example *true* or *false*. In this case f is given by $f : \mathbb{D} \to N \subset \mathbb{N}$. If we want to describe concepts from a terminology T, f is given by $f : \mathbb{D} \to T$. The function may also return words σ^* from a language L which leads to $f : \mathbb{D} \to \Sigma^*$. Here Σ^* denotes the set of all words (or strings)

over a given alphabet or language. We can even consider a subset from the language which leads to $f : \mathbb{D} \to L$. If we do not need a description function we can simply set $f = \text{id}$ to the identity function. As we can see the cut between $\mathbb{X}$ and the other sets is not necessarily empty.

In some cases it is useful to assume that $\mathbb{X} = f(\mathbb{D})$ and thus q as a surjective mapping. It follows that f has a right inverse q with $f \circ q = \text{id}_{\mathbb{X}}$. This function is necessary to model some problems. Usually we cannot assume that f is also an injective mapping: A description set $x \in \mathbb{X}$ may have more than one origin in $\mathbb{D}$.

Considering an element $\mu \in \mathbb{X}$ and $\mathbb{X}$ as the description set of a search engine, R can be explicitly set as $E = q(\mu)$ with the right inverse of f. Here q denotes the search function with $q : \mathbb{X} \to \mathbb{D}$.

D. Evaluation Function

For several problems we need an *evaluation function* $e : \mathbb{E} \to [0, 1]$ which is either a similarity measure sim, an error measure err or a weight $weight$. If it is not applicable, we may use the identity function id. The set $\mathbb{E}$ must be set according to our optimisation goal. If we optimise $\mathbb{D}$, for example by adding new documents or additional information, $\mathbb{E} = \mathbb{D} \times \mathbb{D}$. The same holds, if we want to find an optimal subset $R \subset \mathbb{D}$.

If we want to optimise our description function f, we must use the function space $\mathbb{D}^{\mathbb{X}} = \mathbb{E}$. An evaluation of the reference standard will be even more complex, see below. Then $\mathbb{E} = E$ applies.

E. Reference Standard

Usually the evaluation process cannot be done without an external criterion. In this cases we can add a *reference standard* or *evaluation set*

$$E \subset \left\{ \mathbb{D}^n \times \left(\mathbb{D}^{\mathbb{X}}\right)^m \times \left(\mathbb{D}^{[0,1]}\right)^o \right\}$$

to optimize our result. We either have one single subset of $\mathbb{D}$, or two subsets – a positive and a negative reference standard. We may also have a ranked list of subsets of $\mathbb{D}$. A description function in the function space $\mathbb{D}^{\mathbb{X}}$ or n of them could also be set as a reference standard, as well as one or o evaluation functions out of the function space $\mathbb{D}^{[0,1]}$. This is sometimes denoted as one or many *gold standards*. This can be very complex, but usually problems only need one of these sets.

If not feasible or unused, we may also set $E = \emptyset$.

F. Problem Description

Natural Language Processing problems can thus be described by a five-tuple. We can denote them by a combination p of

$$p = \mathbb{D}|R|f : \mathbb{D} \to \mathbb{X}|e : \mathbb{E} \to [0, 1]|E$$

with a domain set $\mathbb{D}$, a domain subset R, a description set $\mathbb{X}$ and a description function f as well as an evaluation function e evaluating on the set $\mathbb{E}$ according to the reference standard E. We usually have four parameters given and want to obtain an optimal solution for the fifth. The optima result with respect to the problem will be denoted in bold letters. We can add an additional index for ambiguous notations.

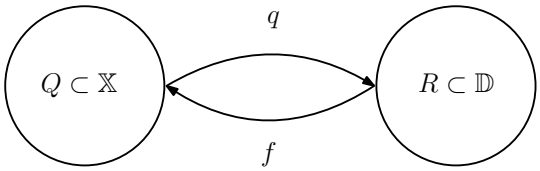
If we have an optimal algorithm we only need one computation step. If we have a heuristic returning approximate values, we may use the output of the first iteration as an input for the next iteration. We will discuss, that similar approaches usually only differ with respect to the chosen set $\mathbb{X}$.

III. Search Queries and Cluster Labels

A. Generating and optimisation of Search Queries

In this paper we use a very generic definition of search engines and search queries. A search engine is a function $q : \mathbb{X} \to \mathbb{D}$ which outputs a set of documents or any other content of the domain set if the input is a subset of a description set $\mathbb{X}$ which we cal search query.

The problem of generating search queries usually has a domain set $\mathbb{D}$ restricted by the database of the search engine. The return value of our problem is a search query $\mu \in \mathbb{X}$ so that $q(\mu) = R$. Thus R is the subset of documents for which we want to create a search query. We have a mapping from



one element in $\mathbb{X}$ to a subset of $\mathbb{D}$. q is thus the right inverse of the description function f and $f \circ q = \text{id}_{\mathbb{X}}$. Not only does f have to be surjective, but we also have to assume that even q is surjective. Every document in the target set $\mathbb{D}$ should be a target of some search query.

It is very easy to see that this is usually not given in reality: Assume q is a websearch, $\mathbb{X}$ the web search description and $\mathbb{D}$ the set of all web pages available. Some of them may not be indexed due to restrictions made to the robots crawling and indexing the web. We can sail around this by restricting $\mathbb{D}$ to $q(\mathbb{X})$. Then f should be the right inverse of q with $q \circ f = \text{id}_{\mathbb{D}}$.

We can also see, that $\forall d \in \mathbb{D}$ several $\mu_1, ..., \mu_n \in \mathbb{X}$ exists with $d \in q(\mu_i)$ – neither q nor f are injective mappings. If we want to find the optimal μ we need to define some sort of metric on elements in $\mathbb{X}$. This can be very complex. If we assume, that we have a terminology T and a simple algebra with $\vee$ and $\wedge$, we can simplify $\mathbb{X} = \mathcal{P}(T, \vee, \wedge)$ and take the length of $\mu \in \mathbb{X}$ as a metric. But if all documents have a unique index stored in $\mathbb{X}$ the shortest search query might consist of a concatenation of these indexes listing all documents in R.

Thus, the simplest evaluation function $e : \mathbb{D} \to [0, 1]$ is set by

$$err_1(d_i, d_j) = \begin{cases} 1 & i \neq j, f(d_i) \neq f(d_j), d_i, d_j \in R \\ 1 & i \neq j, f(d_i) = f(d_j), d_i \text{ or } d_j \in R \\ 0 & \text{else} \end{cases}$$

If f, the description function with the image set $\mathbb{X}$, does not map two documents in R to the same element, which is the

search query $\mu \in \mathbb{X}$, we count an error. Same happens, if another document not in R is mapped to R. Thus we want to find a description function f so that $f(R) = \mu \in \mathbb{X}$ with $q(\mu) = R$. It follows that the problem is given by

$$p = \mathbb{D}|R|\mathbf{X}|err_1|R$$

This is the simplest formulation of the stated problem. As discussed, it can be more complex. We have not defined a proper quality measure for search queries $\mu \in \mathbb{X}$. In addition, the space $\mathbb{X}$ may be very complex and it is not clear, if it is – like $\mathbb{D}$ – a discrete space with a proper metric. In addition, although f is a surjective mapping and q can be set to be surjective, it is left open, if one of these mapping might also be injective.

B. Generating and optimisation of Cluster Labels

A clustering is usually done on a domain set $\mathbb{D}$ and leads to several clusters $C_1, ..., C_n$, $n \in \mathbb{N}$. If $\mathbb{D} = \mathcal{P}(\mathbb{S})$, these clusters are explicitly coded in the set $\mathbb{D}$. Finding cluster labels is the task of assigning a subset of a description set $\mathbb{X}$ with the description function $f : \mathbb{D} \to \mathbb{X}$ to a cluster $R \in \{C_1, ..., C_n\}$. We might consider an evaluation function measuring the distance between the description between two documents in R, $|f(d_i) - f(d_j)|$. But we need to assume a proper metric on $\mathbb{X}$ to do so. This leads to very complex questions. For example: What is a proper metric on a space of boolean algebra? The easiest evaluation function is thus given by

$$err_2(d_i, d_j) = \begin{cases} 1 & i \neq j, f(d_i) \neq f(d_j), d_i, d_j \in R \\ 1 & i \neq j, f(d_i) = f(d_j), d_i \text{ or } d_j \in R \\ 0 & \text{else} \end{cases}$$

Here we define that every two documents in R must share the same cluster labels. This cluster label has to be unique to this cluster. The reference standard can also be set to R. Thus the problem of generating and optimisation of cluster labels is given by

$$p = \mathbb{D}|R|\mathbf{X}|err_2|R$$

where the resulting label set is the image $f(R) \subset X$. Depending on the choice of X this either leads to a set of metadata, terms, sentences or any subset of natural language. Again, this problem can be very complex.

C. Search Queries and Cluster Labels are closely connected

In our introduction we already discussed, that Hagen et al. found out that both problems are similar, see [3]. It is easy to proof that given the same domain set $\mathbb{D}$, image set $\mathbf{X}$ of the description function and the same evaluation function both problems are equivalent. Thus, they are closely connected.

Lemma III.1. *Let $\mathbb{X}$ be a description image set. For every solution f of $p_1 = \mathbb{D}|R|\mathbf{X}|err_1|R$ this is also an optimal solution of $p_2 = \mathbb{D}|R|\mathbf{X}|err_2|R$.*

Proof. This follows directly, since $err_1 = err_2$. □

Same follows directly for the inverse:

Lemma III.2. *Let X be a description image set. For every solution f of $p_2 = \mathbb{D}|R|\mathbf{X}|err_2|R$ this is also an optimal solution of $p_1 = \mathbb{D}|R|\mathbf{X}|err_1|R$.*

Thus both problems are equivalent if we consider the same domain set $\mathbb{D}$, image set X of the description function and the same evaluation function. We can conclude that we can use the same or similar heuristics for solving both problems. Usually a search query language is not used for representing cluster labels. But since query languages and natural languages are not only highly connected but merge more and more (see [9] or [10]) we follow that in future both problems will be even more connected. We will now discuss a small example.

D. Example

We will do a generation and optimization of cluster labels with a similar approach to Borkowski et al. [11], Kanavos et al. [12] and Demner et al. [13]. All of them use a taxonomy of categories like Medical Subhect Headings (MeSH, see https://www.nlm.nih.gov/mesh/) and process the documents using tfidf-method. We will use SCAIView, see [14] or https://www.scaiview.com), an information retrieval system for knowledge discovery for a similar approach. SCAIView was used in many recent research projects, for example regarding neurodegenerative diseases [15], brain imaging features [16] and other theoretic research like document clustering, see [17]. The advantage is, that SCAIView already provides us with Named Entities for MeSH but also other ontological representing biomedical entities. Thus we get a better coverage of text with named entities.

Our domain set $\mathbb{D}$ is MEDLINE data, and R a subsets of MEDLINE data. MEDLINE (Medical Literature Analysis and Retrieval System Online) is a bibliographic database maintained by the National Center for Biotechnology Information and covers a large number of scientific publications from medicine, psychology, and the health system. For the clustering use case, we study MEDLINE abstracts and associated metadata that are processed by ProMiner, a named entity recognition system, see [18], and indexed by the semantic information retrieval platform SCAIView.

Our goal is, to find a unique representation of R in $\mathbb{X}$. Let $f(d) = \mu$ for all $d \in R$. We have to define the description set $\mathbb{X}$.

Borkowski et al. [11] processed a ranked list of categories with their weights. We will follow Kanavos et al. [12] and use all ontologies available at SCAIView. To make our approach easier, we will limit our image set $\mathbb{X}$. Let $\mathbb{X}$ be the SCAIView search query set limited to NE.

R is the document set retrieved by a list of pubmed identifiers. In our example, we have R as the result of a list of 38 PMIDs. $\mathbb{D}$ is the set of all documents in SCAIView databse. Querying the Lucene backend we find a list of 654 NE $N = n_1, ..., n_{654}$ and the documents containing them, which we donate by $l(n_i)$. The list $n_i : l(n_i)$ has the form

```
CHEBI:36929 : [38]
MESH:E05.598.500 : [9, 11, 18, 36]
ENTREZGENE:3630 : [29, 37]
MESH:C10.597.742 : [9]
ENTREZGENE:387244 : [34]
MESH:C10.597.622 : [14]
MESH:F03.900.675.400 : [12]
MGI:96543 : [6, 15, 29, 32, 33]
CHEBI:6271 : [26]
...
```

We will now use a novel set covering approach. Following [12] the labels for distinct subsets can be seen as potential candidates for cluster labels. For example we can cover r with n terms:

```
1  ;MESH:F01.700.039
1  ;MESH:C10.597.606.057
1  ;SWISSPROT:HUMAN
1  ;MESH:D059445
1  ;MESH:C23.888.592.604.039
2  ;CHEBI:36929  ;MESH:F01.700.039
2  ;CHEBI:36929  ;MESH:C10.597.606.057
...
```

If we use sim_1 as an evaluation measure, we are nearly done. But we might get more documents in $\mathbb{D}$ by querying these labels.

We can now construct a hierarchical tree using the logical operators *and* and *or* in $\mathbb{X}$. We will do this by considering a graph $G = (V, E)$ with nodes $V = N$, in our example $V = \{n_1, ..., n_{654}\}$. We add weighted edges between two nodes n_i, n_j if $l(n_j) \subset l(n_i)$. The weight is set to zero if $\nexists n_k \in N$ such that $l(n_j) \subset l(n_k) \subset l(n_i)$. Otherwise we set the weight $w(n_i, n_j)$ to the largest number $\mathfrak{w}$ so that $\mathfrak{w}$ elements in N exist with $l(n_j) \subset l(n_1) \subset ... \subset l(n_{\mathfrak{w}}) \subset l(n_i)$. Thus the weight is zero if the document set is a direct subset of the other document set. Otherwise, it is the largest number of subsets that lie in between. Finding the minimum spanning tree(s) in this graph G and connecting all nodes with `AND` and the `OR` of their child nodes lead to the solution μ:

```
MESH:F01.700.039
AND MESH:C10.597.606.057
AND SWISSPROT:HUMAN
AND MESH:D059445
AND MESH:C23.888.592.604.039
AND (
  MESH:E05.598.500 AND (MGI:95574 OR MESH:C10
      .597.742)
  OR ENTREZGENE:3630 AND (MGI:96542 OR MESH:C19
      .246.300)
  OR
  ...)
```

The description function f is a heuristic finding one spanning tree on G with the named entities covering the domain subset R.

This is both: a correct solution of clustering labeling of R on $\mathbb{X}$ obtained by f as well as a possible solution of a search query so that $q(\mu) = R$.

As we can see, even this simple approach needs a complex heuristic. Although finding minimum spanning trees is usually in $\mathcal{FP}$, we can construct more complex examples that are $\mathcal{NP}$-complete. It would be very beneficial to find problems that are in $\mathcal{P}$.

This approach can now very easily be transferred into natural language, although a very complex boolean algebra might not be very helpful to human readers. This leads to another problem we already discussed: How is the space $\mathbb{X}$ defined? Is it a metric space? Is q an injective mapping? This reformulation of both problems is very helpful to discuss and proof the complexity and the real underlying problems and to find more suitable heuristics and algorithms. But it also leads to new questions and problems.

IV. More information extraction problems

Information extraction problems can be transferred into $p = \mathbb{D}|R|\mathbb{X}|sim|\emptyset$ with a result information description image set $\mathbb{X}$ for R. Here f is a function that extracts some information out of a document d: This may be natural language $f : \mathbb{D} \to L$ or another subset of $\mathbb{D}$ or a mapping to ontologies or terminologies.

We will discuss some examples and point out the benefits of our new approach.

A. Named Entity Recognition

Named Entity Recognition (NER) was initial proposed as the task to identify names, location and temporal constructs in text [19]. Over the decades this initial definition expanded to detect arbitrary concepts, defined as things of thought [20]. Different Algorithms developed and adapted to NER over time from simple gazetters [21] over rule based engines [18] and probabilistic context-free grammar [22] to Conditional Random Field [23] and neuronal networks [24]. No matter on how the algorithm solves the problem in the end it needs to link a sequence of character to one or many concepts. Therefor a model, a function, is constructed that encode how this should happen. This model is either generated by manual labour or my machine learning approaches or mixtures in between.

The evaluation of NER applications is often done by comparing an obtained result to a reference (gold) standard [25]. For this comparison a function $e : \{\mathbb{S} \times [0, 1]\} \times \{\mathbb{S} \times [0, 1]\} \to \{0, 1\}$ is needed that assesses if a Named Entity (NE), a Concept, is correctly detected or not. Based on the discrete values of the function Precision, Recall and a F-score are computed and are used as a performance indicator [25]. For the definition of e we assume that the target set of the description function, performing the NER, matches the definition of a reference standard. This allows to use a result of a description function a as a reference for a different function.

As initial mentioned NER is the task to link a sequence of character to concepts. Subsequent the description function looks like $f : \Sigma^* \to Concept$. If we assume a Concept is encoded as an SDA the function is a mapping from a sequence of character to SDAs. We also encode sequences of characters as SDAs so the function can be refined as $f : \mathbb{S} \to \mathbb{S}$. To also encode the uncertainty of such a mapping the final function is $f : \mathbb{S} \to \{\mathbb{S} \times [0, 1]\}$. To fit into the proposed schema, the

function needs to map from a document $\mathbb{D}$, thus we define $\mathbb{D} = \mathcal{P}(\mathbb{S})$ and $f : \mathbb{D} \to \{\mathbb{S} \times [0, 1]\}$

We transferred the application $p = \mathbb{D}|\emptyset|f|\emptyset|\emptyset$ and evaluation $p = \mathbb{D}|\emptyset|f|e|\{\mathbb{S}\times[0, 1]\}$ phase of NER into the initial proposed five-tuple. Likewise we can decode a learning phase, therefore we define a training set $T = \{\mathbb{S}^1 \times [0, 1]\}, \mathbb{S}^1 \subset \mathbb{D}$. The function f than can be learned resp. optimized by using the training set T and the evaluation function e. Subsequent a learning phase is expressed as $p = \mathbb{D}|T|\mathbf{f}|e|\emptyset$. Combing all three phases the initial proposed five tuple is constructed:

$$p = \mathbb{D}|T|\mathbf{f}|e|\{\mathbb{S} \times [0, 1]\}$$

We like to illustrate this with a gazatter based approach. We choose gazetter as an example because it is simple to understand and it eases discussion about extending to more advanced methods. Assume we have a gazetter, a list, with n Concepts $g = g_1, g_2, ..., g_n$. Each Concept $g_i = \{g_{i1}, g_{i2}, g_{im_i}\}$ is a set of alternative names (pairwise disjoint), where g_{i1} is the Representative. The description function f maps SDAs that encode sequences of character to a SDA that encode the Representative of a Concept. For a simple gazetter the function could look like the function below where an exact string match [26] between the character sequence and an alternative name is required.

$$f(n) = \begin{cases} \{S(g_{11}), 1\} & \text{if } \exists g_{1j} = n, 1 \leq j \leq m_1 \\ \{S(g_{21}), 1\} & \text{if } \exists g_{2j} = n, 1 \leq j \leq m_2 \\ \vdots & \vdots \\ \{S(g_{n1}), 1\} & \text{if } \exists g_{nj} = n, 1 \leq j \leq m_n \end{cases}$$

This function can be extend to a fuzzy string matching [27]. The fuzziness can be encoded via a normalized edit distance [27] in the second argument. Orthogonal an extension on the used data is possible. The function could work on Token , Stems or Lemmas [25] instead of character sequences. This requires a preprocessing of the gazetter as well as a transformation of SDA. Or alternatively a minor refinement of the tuple, switching from character SDAs to e.g. Token SDAs. However this is possible within the proposed five tuple and shows the generality of our systematization. The same transformation approach can be used to directly incorporate various machine learning methods, like [23], [24], [22]. The SDA, of the training set can be decoded into a better suited feature representation for the used method and the results can also be transformed into SDAs. Alternatively the method could directly use SDA as features.

As it can be see the systematization is a common base for various methods. Various methods can be transferred into this tuple representation by a transforming the data from and into an SDA. This shows the strength and flexibility of SDAs and the proposed tuple.

B. Text summarization

Text summarization is the task of assigning a short summary in natural language L to a document $d \in \mathbb{D}$. Thus our description set $\mathbb{X} = L$ and the complete problem has the form

$$p = \mathbb{D}|R|\mathbf{L}|sim|\emptyset$$

with a result information description $f(R)$ for R. Here f is a function that extracts some information in form of language out of a document d: $f : \mathbb{D} \to L$. Once again we have to ask how our evaluation function sim works. Is sim just the vector distance in a vector-space representation of L? If we limit L to a list of terms, a terminology or ontology summarising the document, this might be suitable. But considering the context of text might be more helpful. We can find several examples in literature: Demner et al. [13], who generated extractive summaries for abstracts of documents in MEDLINE. Barzilay et al. [28] did use lexical chains, without considering the semantic interpretation. Gong et al. [29] rather explicitly considered the semantic of the texts.

Thus all these approaches differ in the definition of the domain set $\mathbb{D}$. It may contain a simple list of texts or abstracts, but it may as well contain semantic digital assets $\mathbb{S}$ considering the semantics and context. In addition the description function is another criterion for distinction. We may add them as additional index to $\mathbb{X}$. This leads to $p = \mathbb{D}|R|\mathbf{L}_{lexical\_chains}|sim|\emptyset$ for [28] or $p = \mathbb{S}|R|\mathbf{L}_{semantics}|sim|\emptyset$ for [29].

This novel problem description provides a helpful framework for sorting the approaches found in literature.

C. Relation Extraction

The task of extracting relational facts – or relations – from a text is the combination of two or multiple entities in a computable format. This is usually either done by manual curation (see [30]) or by supervised or unsupervised learning (see [31] or [32]). Relation extraction is widely used in biomedical research, biology or toxicology to handle the growth of publications and data available. In addition it is used in medical research, see [33]. After relation extraction computable networks are created, see [32].

Let $\mathbb{D} = \mathcal{P}(\mathbb{S})$ be a set of documents denoted by SDAs. Our description set contains all relations between SDAs. Thus it is the set of all functions from SDAs to SDAs: $\mathbb{X} = \mathbb{D}^{\mathbb{D}}$. In addition we need a similarity measure that maps relations from documents to SDA terms:

$$sim : \mathbb{X} \to [0, 1]$$

Then doing Relation Extraction is the task of finding an optimal description function

$$f : \mathbb{D} \to \mathbb{X}$$

that either maps SDAs for sentences to one or more relations between SDAs or to 0 if a SDA has no NE. Thus we find

$$p = \mathbb{D}|R|\mathbb{D}^{\mathbb{D}}|sim|\emptyset$$

A lot of question have to be left open: How can we define f according to the solutions found in literature? Could $\mathbb{X}$ be reduced to a subset without loosing information? Once again a systematization of approaches found in literature could be made, although this will be part of future work.

V. Discussion and further research

We proposed a novel formal schema for information retrieval and natural language processing problems and reformulated several well-known problems. Our schema is helpful to sort NLP-problems according to their underlying and inherent structure and to identify the complex parts to solve the problem.

Discussing the equivalence between cluster labelling and finding search query we proofed that they are – obviously – equivalent if they share the same description set and the same evaluation function. This directly leads to the conclusion, that most NLP-problems have a core problem that can be solved with distinct heuristics and algorithms. Finding an evaluation function and a description set is not a core problem of computer science, but deeply related to linguistics and applied computer science. Our new approach will help to group problems and foster synergies for optimization and offer a better description with terms of theoretical computer science. Here we already reduced a simplified search query problem to a graph problem.

We left several open questions. Further research has to be done with focus on time and space complexity – what is the computational complexity in these natural language problems? Here the integration of formal language theory will be the next step. Also unsupervised and supervised learning can be expressed with our novel approach, more research has to be done regarding this. In addition, our paper is based on text data. But we can also express binary data such as speech and images in $\mathbb{D}$.

In this paper we could only discuss some early work on the preliminaries and provide a few short examples. We hope that the impact of our schema is a better categorization of NLP-problems and a better communication between application and theoretical informatics, leading to more efficient algorithms and heuristics.

References

[1] C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[2] A. Clark, C. Fox, and S. Lappin, *The handbook of computational linguistics and natural language processing*. John Wiley & Sons, 2013.

[3] M. Hagen, M. Michel, and B. Stein, "What was the query? generating queries for document sets with applications in cluster labeling," in *International Conference on Applications of Natural Language to Information Systems*. Springer, 2015, pp. 124–133.

[4] D. Babeanu, A. A. Gavrila, and V. Mares, "Strategic Outlines: Between Value And Digital Assets Management," *Annales Universitatis Apulensis: Series Oeconomica*, vol. 11, no. 1, p. 318, 2009.

[5] J. P. Hopkins, "Afterlife in the Cloud: Managing a Digital Estate," *Hastings Science and Technology Law Journal*, vol. 5, p. 209, 2013.

[6] H. Malissa, "Automation in und mit der Analytischen Chemie IV," *Fresenius' Zeitschrift für analytische Chemie*, vol. 256, no. 1, pp. 7–14, Feb. 1971.

[7] M. Jacobs, S. Hodapp, and J. Dörpinghaus, "SDA: Towards a novel Knowledge Discovery Model for Information Systems," in *Proceedings of the 11th IADIS International Conference Information Systems 2018*. IADIS, 2018, pp. 300–302.

[8] J. Dörpinghaus, M. Jacobs, and J. Fluck, "Graph based Discovery in biomedical Information Systems connecting scientific Texts with structured Expoert Knowledge," in *Proceedings of the 11th IADIS International Conference Information Systems 2018*. IADIS, 2018, pp. 297–299.

[9] D. Suryanarayana, S. M. Hussain, P. Kanakam, and S. Gupta, "Natural language query to formal syntax for querying semantic web documents," in *Progress in Advanced Computing and Intelligent Engineering*. Springer, 2018, pp. 631–637.

[10] D. Melo, I. P. Rodrigues, and V. B. Nogueira, "Semantic web search through natural language dialogues," in *Innovations, Developments, and Applications of Semantic Web and Information Systems*. IGI Global, 2018, pp. 329–349.

[11] P. Borkowski, K. Ciesielski, and M. A. Kłopotek, "Semantic classifier approach to document classification," *arXiv preprint arXiv:1701.04292*, 2017.

[12] A. Kanavos, C. Makris, and E. Theodoridis, "Topic categorization of biomedical abstracts," *International Journal on Artificial Intelligence Tools*, vol. 24, no. 01, p. 1540004, 2015.

[13] D. Demner-Fushman and J. Lin, "Answer extraction, semantic clustering, and extractive summarization for clinical question answering," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 841–848.

[14] E. Younesi, L. Toldo, B. Müller, C. M. Friedrich, N. Novac, A. Scheer, M. Hofmann-Apitius, and J. Fluck, "Mining biomarker information in biomedical literature," *BMC medical informatics and decision making*, vol. 12, no. 1, p. 148, 2012.

[15] M. A. E. K. Emon, R. Karki, E. Younesi, M. Hofmann-Apitius *et al.*, "Using drugs as molecular probes: A computational chemical biology approach in neurodegenerative diseases," *Journal of Alzheimer's Disease*, vol. 56, no. 2, pp. 677–686, 2017.

[16] A. Iyappan, E. Younesi, A. Redolfi, H. Vrooman, S. Khanna, G. B. Frisoni, and M. Hofmann-Apitius, "Neuroimaging feature terminology: A controlled terminology for the annotation of brain imaging features," *Journal of Alzheimer's Disease*, vol. 59, no. 4, pp. 1153–1169, 2017.

[17] J. Dörpinghaus, S. Schaaf, J. Fluck, and M. Jacobs, "Document clustering using a graph covering with pseudostable sets," in *Computer Science and Information Systems (FedCSIS), 2017 Federated Conference on*. IEEE, 2017, pp. 329–338.

[18] D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck, "ProMiner: rule-based protein and gene entity recognition." *BMC bioinformatics*, vol. 6 Suppl 1, p. S14, 2005.

[19] R. Grishman and B. Sundheim, "Message understanding conference-6: A brief history," in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, vol. 1, 1996.

[20] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[21] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, ser. CoNLL '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 147–155. [Online]. Available: http://dl.acm.org/citation.cfm?id=1596374.1596399

[22] J. R. Finkel, A. Kleeman, and C. D. Manning, "Efficient, feature-based, conditional random field parsing," *Proceedings of ACL-08: HLT*, pp. 959–967, 2008.

[23] R. Klinger, C. M. Friedrich, J. Fluck, and M. Hofmann-Apitius, "Named entity recognition with combinations of conditional random fields," in *Proceedings of the second biocreative challenge evaluation workshop*, 2007.

[24] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[25] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press, 2008, vol. 39.

[26] C. Charras and T. Lecroq, *Handbook of exact string matching algorithms*. Citeseer, 2004.

[27] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

[28] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," *Advances in automatic text summarization*, pp. 111–121, 1999.

[29] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2001, pp. 19–25.

[30] J. Fluck, S. Madan, S. Ansari *et al.*, "Belief-a semiautomatic workflow for bel network creation," in *Proc. 6th Int. Symp. Semant. Min. Biomed*, 2014, pp. 109–113.

[31] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1003–1011.

[32] J. Fluck, S. Madan, S. Ansari, A. T. Kodamullil, R. Karki, M. Rastegar-Mojarad, N. L. Catlett, W. Hayes, J. Szostak, J. Hoeng *et al.*, "Training and evaluation corpora for the extraction of causal relationships encoded in biological expression language (bel)," *Database*, vol. 2016, p. baw113, 2016.

[33] F. Rinaldi, T. R. Ellendorff, S. Madan, S. Clematide, A. Van der Lek, T. Mevissen, and J. Fluck, "Biocreative v track 4: a shared task for the extraction of causal network information using the biological expression language," *Database*, vol. 2016, 2016.