Bauhaus-Universität Weimar
Faculty of Media
Computer Science and Media

# Constraints in Web People Search

Bachelor Thesis

Johannes Kiesel

1. Supervisor: Prof. Dr. Benno Stein
2. Supervisor: Prof. Dr. Hagen Höpfner

Date of submission: 2.4.2012

# Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Weimar, 2.4.2012

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Johannes Kiesel

## Abstract

Retrieving information on certain people seems to be a simple task when considering the amount of personal information which is available on the Internet. It is, however, difficult to describe a person in detail in order to filter out all information about other individuals with the same name, but broad enough to avoid the filtering of information about the individual in question. In web people search it is attempted to identify all web pages which possibly concern the individual of interest and to group the web pages by the referents they concern. Moreover, personal information in the form of attributes is extracted for each referent from the different web pages. Then, the task of retrieving information on a certain person reduces to selecting the individual in question from a list of people, each of whom is described by personal attributes.

In this thesis, we introduce a framework for the representation of personal information as instance-based constraints, which then serve as guidance for the clustering of the different referents. We detail the challenges which arise from the employment of uncertain information from web pages in decision making and propose some solutions to them. More specifically, we employ and introduce methods for the matching of personal information (*exact* and *soft*), the addition of constraint strengths (*maximum* and *multiplication*), and the resolution of conflicts within the resulting constraint sets (*taking-must-link-constraints* and *raising-threshold*). Furthermore, we propose two statistical properties of person attributes, both of which can serve as indicators for the suitability of a person attribute for constraint generation: the *referents-per-value* and the *values-per-referent* ratio.

Finally, we conduct a series of experiments to compare the performance of the different methods within the context of web people search. We employ the corpus of the second workshop on web people search (WePS-2) in our experiments. The workshop consisted of both a web page clustering and an person attribute extraction task. Instead of extracting the person attributes by ourselves, we apply either the gold standard of the WePS-2 workshop, or the attribute values extracted by the best participating team. We show that our proposed framework is comparable to state-of-the-art web people search algorithms if person attribute values of the quality of human extracted ones are supplied.

# Acknowledgements

So it has come to this. I first want to apologize that this page had not made it into the versions of this thesis which I have already submitted. After finishing the chapters there was just too little time left for an appropriate formulation of my thanks. I am glad to be able to now rectify this omission of mine.

Although the extracted person attribute values by Chen et al. are only used in the last experiments of this thesis, an important part would be missing without them. Thanks to the kindness of Ying Chen I was able to concentrate on the use of the person attribute values and nevertheless analyze the proposed framework in the case of algorithmically extracted values.

English is not my mother tongue and grammatical rules have always been mysterious and cryptic principles for me. If I have been able to state my points clearly, it has been so to the merit of Mr. Atkinson and my tutor Dennis. I especially like to thank Dennis for the fruitful discussions on the structure of the single paragraphs, the sections, the chapters, and also the whole thesis.

I am grateful to the whole Computer Science and Media staff for their efforts on providing a comfortable environment to study in. Especially I would like to thank Mr. Stein, the head of the Webis group which I am thankful to be a student assistant in. I would also like to give thanks to my fellow students for enlightening conversations as well as the whole non-science-stuff of the past three and a half years.

I am grateful to my family for their support over all the years which I have seen so far. Especially for their enthusiasm for the little studies and projects in which I participated—including this one. And the last “thank you” is for Dora. For patience and for being there. And for some other things.

I really look forward to the next years here at the Bauhaus Universität in which I will continue my studies together with most of you. Years of studying, living and moving the

*rocks on the road.*

# Contents

# Chapter 1

# Introduction

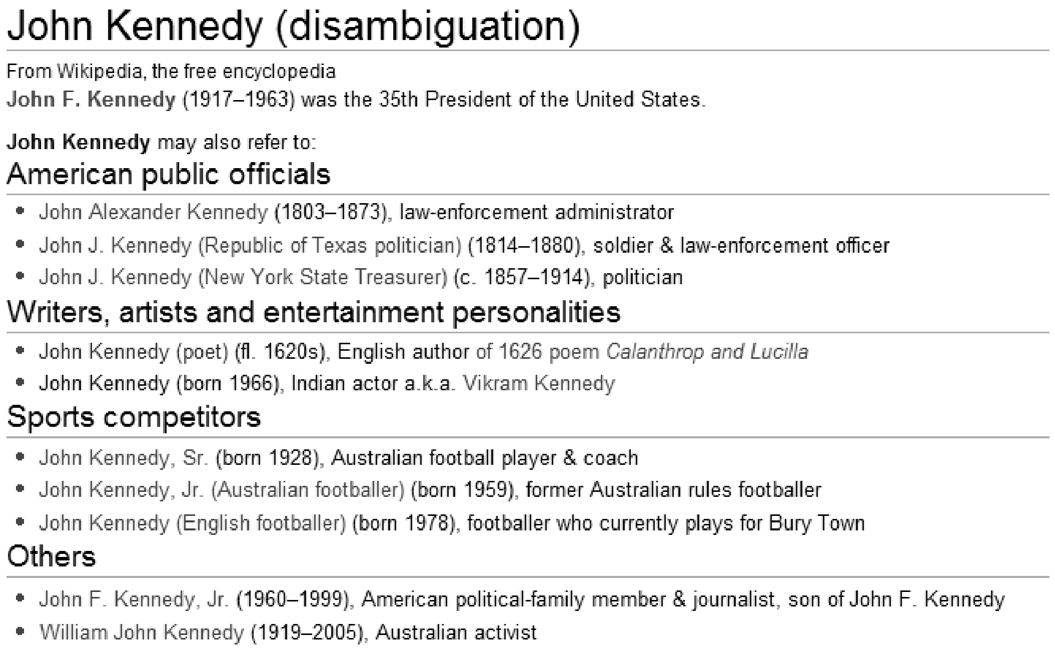
Web search engines allow the user to explore the Internet by returning web pages relevant to the queries issued by the user. In order to help a user to decide which web page to visit, short text snippets summarizing the content of each result page are shown. By revealing the context in which the search terms occur, these snippets allow the user to judge the relevance of each web page.

*Web people search* (i.e., searching for information about a specific person on the Internet) requires additional methods to help the user identify relevant web pages. An intuitive method of searching for information is to use the name of the specific person (the *target name*) as a query to a search engine. Person names are, however, shared among several people; consequently, the results returned will contain web pages referring to several different individuals. Moreover, if there is a famous person with the target name, it is likely that most of the result web pages will refer to the famous person. If the user is only interested for this famous person, using the search engine is a reasonable approach. Alternatively, neutral information can also be found in online encyclopedias such as *Wikipedia.* According to a study of Spink et al. [2004], however, only 25% of person name queries, which equals 1% of all queries, contain the name of a famous person. But if the user is looking for a non-famous person, the web pages retrieved by the search engine are dominated by web pages referring to various people of the same name.[1] Furthermore, it is unlikely that an article about the person of interest exists in Wikipedia.

Over the few last years, several people search engines have been released;[2]

[1] An additional term to further specify the person (e.g., the city the person lives in) occurs only in 20% of person name queries [Spink et al., 2004].

[2] For example *ZoomInfo* (`http://www.zoominfo.com`), which specializes in business-people, or *ArnetMiner* (`http://www.arnetminer.org`), which searches for academic researchers.


John Kennedy (disambiguation)

From Wikipedia, the free encyclopedia

**John F. Kennedy** (1917–1963) was the 35th President of the United States.

**John Kennedy** may also refer to:

American public officials

- John Alexander Kennedy (1803–1873), law-enforcement administrator
- John J. Kennedy (Republic of Texas politician) (1814–1880), soldier & law-enforcement officer
- John J. Kennedy (New York State Treasurer) (c. 1857–1914), politician

Writers, artists and entertainment personalities

- John Kennedy (poet) (fl. 1620s), English author of 1626 poem *Calanthrop and Lucilla*
- John Kennedy (born 1966), Indian actor a.k.a. Vikram Kennedy

Sports competitors

- John Kennedy, Sr. (born 1928), Australian football player & coach
- John Kennedy, Jr. (Australian footballer) (born 1959), former Australian rules footballer
- John Kennedy (English footballer) (born 1978), footballer who currently plays for Bury Town

Others

- John F. Kennedy, Jr. (1960–1999), American political-family member & journalist, son of John F. Kennedy
- William John Kennedy (1919–2005), Australian activist

**Figure 1.1:** Abbreviated list of Wikipedia's disambiguation page for the name "John Kennedy". The original list contained 54 different people (accessed January 20, 2012).

If a people search engine is queried with a person name, it shows the user a list of people—not web pages—with the given target name. Each person in the list is distinguished from the others by *person attributes* (lifetime, location, occupation, etc.), which have been gathered from the Internet (e.g., social-network profiles or homepages). In fact, a rather similar approach to the problem of person-name ambiguity is taken by Wikipedia; the user can pick the intended person from a handmade disambiguation list (cf. Figure 1.1). In order to automatically generate such a listing of people from web pages, a method is needed that identifies which web pages refer to the same person. This task is also known as *entity-based cross-document coreferencing* [Bagga and Baldwin, 1998]. At the time of writing, proposed methods for coreferencing mainly apply unsupervised clustering techniques [Artiles et al., 2007, 2009, 2010].

In this thesis, the capabilities of person attributes for unsupervised cross-document coreferencing are analyzed. The personal attributes of the individuals found, which are extracted from the web pages, help the user find the intended person. Furthermore, the attributes can be used as clues for entity coreferencing; if, for example, two different web pages refer to a poet "John Kennedy", they are likely to concern the same person. This idea is also used by other web people search systems [Jiang et al., 2010, Wan et al., 2005].

These systems, however, only note that the use of attributes can increase clustering quality. In contrast, this thesis carries out a thorough analysis of the effectiveness of integrating attribute information into the clustering process. Additionally, we provide reasons for the different levels of effectiveness of different person attributes. Moreover, information which suggests that two web pages do *not* refer to the same person is also discussed.

To integrate the attribute information into the clustering process, it is modeled in the form of constraints between two web pages: *must-link* constraints, which state that the two web pages refer to the same referent; and *cannot-link* constraints, which state that the web pages refer to different referents [Wagstaff and Cardie, 2000]. The use of heuristic constraints for clustering is explored in this thesis. Constraints generated from extracted attribute values can be incorrect for three reasons: (1) one of the web pages may contain factual errors; (2) attribute values may have been incorrectly extracted; (3) the rule for constraint generation may be incorrect. In this regard, the situation shown in Figure 1.1 should be considered. The assumption that all web pages which concern a footballer "John Kennedy" refer to the same person would introduce errors into the disambiguation process. The listed footballers can, however, be distinguished by their different dates of birth or (partially) by their nationality. In order to model this relationship, confidence factors are introduced in this thesis. These factors imply an ordering of the attributes. For example, we have more confidence in a difference in date of birth than in a matching occupation. Nevertheless, often multiple clues for the grouping as well as for the separation of a pair of web pages exist. Therefore, we introduce methods of constraint addition, which employ the confidence factors to weigh out the different clues. Furthermore, in order to account for typographical errors and variant spellings in attribute values, methods of partial string-matching are integrated into the constraint generation process.

The constraints are then enforced by the application of a constrained clustering algorithm. The constrained variants of two common clustering algorithms are employed in our experiments: constrained single pass clustering and constrained single-link hierarchical agglomerative clustering. At each step, these algorithms enforce that no constraints are violated. Additionally, in order to compare the generated constraints independent of any clustering algorithm, we propose measures of constraint-set quality (precision and recall).

This thesis is structured as follows. After a discussion of methods for web people search in Chapter 2, problems arising from the use of person attributes as constraints and our proposed solutions to them are detailed in Chapter 3. The proposed methods are then evaluated and the results discussed in Chapter 4. Finally, some concluding remarks are made and an outlook for future work is given in Chapter 5.

# Chapter 2

# Web People Search

Web people search consists of four steps (cf. Figure 2.1). It is initiated by a query containing a person name (*target name*). The objective in web people search is then to identify the individuals sharing the target name (*referents*), and to provide the user with information on the person of interest (*target referent*). At first, relevant web pages to the given query are retrieved. Second, the web pages which concern the same referent are grouped together. Since the referents are not known a priori, a *clustering algorithm* is employed for grouping the web pages by the referent they concern. This step is detailed in Section 2.1. Third, *person attributes*, which assist the user in distinguishing the referents, are extracted from each group (cf. Section 2.2). Measures for evaluating clustering and attribute extraction algorithms are presented in Section 2.1.4 and Section 2.2.1 respectively. As a last step, the groups and person attributes are displayed to the user.

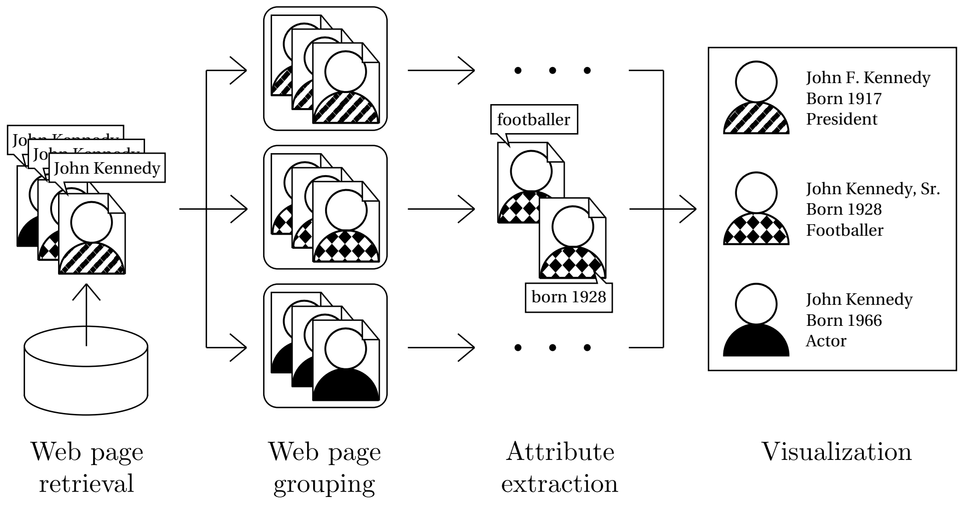


**Figure 2.1:** The four steps in web people search for the query "John Kennedy".

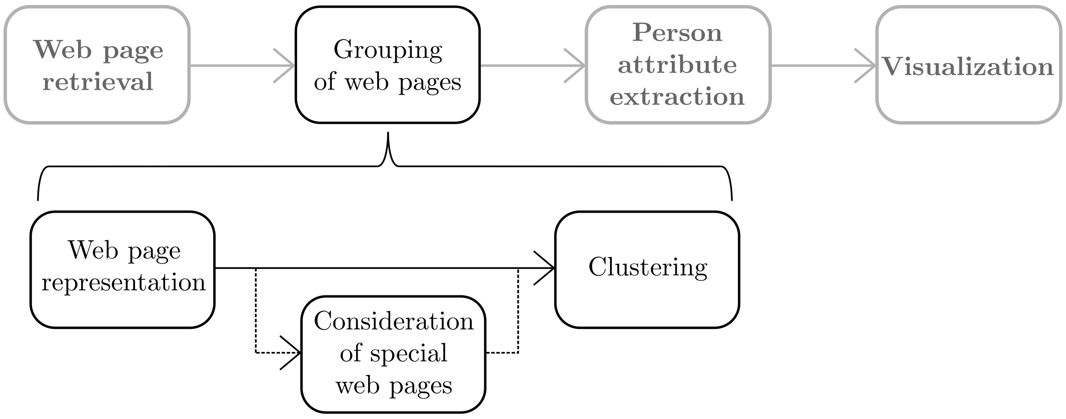


**Figure 2.2:** The four steps of web people search with the clustering step highlighted. Special web pages (i.e., multiple-referent pages and noise pages) can be either identified before or during clustering.

## 2.1 Web Page Grouping

The task of web page clustering can be stated as follows. Given a list of (ranked) web pages $D = d_1, \ldots, d_n$ with each $d_i$ concerning one or more of a set of (unknown) referents $R = \{r_1, \ldots, r_m\}$ with the same target name.[1] Then a partition of $D$ into a set of clusters $\mathcal{C} = \{C_1, \ldots, C_k\}$ with $C_i \subseteq D$ for all $C_i$, has to be specified; ideally, such that the assignment of web pages to clusters corresponds to the (unknown) association of web pages with referents. As a result, each web page within such a cluster concerns the same referent. Since the number of referents $m$ is unknown, $m \neq k$ is often the case.

Instead of identifying the referents ("This page refers to John F. Kennedy, the 35th President of the United States"), clustering aims at grouping the web pages concerning the same referent. The clustering approach to web page grouping, as shown in Figure 2.2, is detailed in the next sections. In a first step, the web pages are represented under a retrieval model, which is detailed in Section 2.1.1. The web pages are then grouped using their representation and a clustering algorithm (cf. Section 2.1.2). As an optional step, special web pages—which concern multiple referents or no referent at all—are considered before the clustering (cf. Section 2.1.3). This means, that web pages concerning multiple referents are split into the parts concerning the referents, and web pages concerning no referent are discarded. This step is optional, since some cluster algorithms are actually able to assign a web page to multiple clusters (and therefore multiple referents), or discard web pages.

[1]Common clustering notation with $d$ being a document.

### 2.1.1 Representation of Web Pages

The textual content of the web pages is the central source of information for web people search systems. Other web page features include the title or the URL [Elmacioglu et al., 2007], but are not employed as often as the text of the web page. Clustering algorithms, however, are not able to group the text as such. Instead, a retrieval model is needed which allows for the computation of pairwise similarities of web pages.

#### The Vector Space Model

In the web people search community, the *vector space model* [Manning et al., 2008] is often chosen for web page representation [Artiles et al., 2009]. The representation bases on the assumption, that each document can be described by the terms it contains. A document is represented as a vector as follows. Let $T$ be a list of terms with $T = t_1, \ldots, t_l$. Each web page $d$ of a (ranked) list of web pages $D = d_1, \ldots, d_n$ is then represented as a $l$-dimensional vector $\mathbf{d}$ in a common vector space. To this end, the number of occurrences of each term $t$ of $T$ in the textual content of each web page $d$ is determined; The number of occurrences of a term $t_j$ in a web page $d_i$ is then employed for the calculation of the $j$-th coordinate of the vector $\mathbf{d}_i$. The calculation of this term-document relevance weight $w_{i,j}$ is detailed below. In the simplest case, $T$ is the list of all terms which occur in at least one of the web pages in $D$. As a result, the list of web pages $D$ is mapped onto the list of vector representations $\mathbf{D}$, with $\mathbf{d}_i = (w_{i,1}, \ldots, w_{i,l})$ for each $\mathbf{d}$ in $\mathbf{D}$. If the above mentioned assumption, that a document can be represented by the terms it contains, holds, then similar vectors will correspond to pages with similar content.

#### Term-document Relevance Weighting

To judge the relevance $w_{i,j}$ of the term $t_j$ to the web page $d_i$, often the tf·idf weighting scheme [Manning et al., 2008] is applied. It consists of the *term frequency* (tf) and the *inverse document frequency* (idf). By employing the term frequency, the relevance weight increases with the number of occurrences of the term in the web page; the term frequency $\mathrm{tf}(t, d)$ is defined as the number of occurrences of $t$ in $d$. It is often combined with the inverse document frequency to account for the frequency of terms. The reasoning behind this is, that terms which occur frequently and in most web pages are not well suited for distinguishing web pages, as they are less likely to be related to the web page content. Thus, the more frequently a term occurs in total, the more it has to occur in a web page to be deemed relevant to it. This is expressed by the inverse document frequency:

$$\text{idf}(t) = \log\left(\frac{|D|}{\text{df}(t)}\right)$$

where the document frequency $\text{df}(t)$ is defined as *the number of documents in $D$ that contain $t$.* To account for collections of different sizes, $|D|$, which denotes the number of pages in the web page collection $D$, is utilized. Finally, the two parts are combined:

$$w_{i,j} = \text{tf}{\cdot}\text{idf}(t_j, d_i) = \text{tf}(t_j, d_i) \cdot \text{idf}(t_j) \tag{2.1}$$

It should be noted that not the entire set of terms which occur in the $D$ is used for representation. For the same reason as for using the inverse document frequency, *stopwords* ("and", "the", etc.) are frequently ignored. Furthermore, terms are often reduced to their stem (*stemmed*). Thus, the term relevance becomes independent of the word form in which the term occurs. This is advantageous, as it is widely believed that the word form contains no descriptive information about the content [Porter, 1997]. If only English web pages are considered, the algorithm of Porter [1997] can be employed, which applies a set of rules to chop of the affixes of the terms.

Other methods to improve the representation do not rely on the single terms, but, for example, on word bigrams [Chen et al., 2009], named entities and compound keywords [Ikeda et al., 2009], or Wikipedia concepts [Long and Shi, 2010].

### Web Page Similarity

For web pages represented in the vector space model, the *cosine similarity* [Manning et al., 2008] can be applied as a measure of similarity. The cosine similarity is the cosine of the angle between two vectors $\mathbf{d}, \mathbf{d}'$:

$$\varphi_{\cos}(\mathbf{d}, \mathbf{d}') = \frac{\mathbf{d} \cdot \mathbf{d}'}{\|\mathbf{d}\| \cdot \|\mathbf{d}'\|} \tag{2.2}$$

where $\|\mathbf{d}\|$ denotes the euclidean norm of $\mathbf{d}$.[2] Since each coordinate of the vectors is positive, the similarity is between 0 (very dissimilar) and 1 (virtually identical web pages). Therefore, the similarity depends not on the total number of terms in the web page, but on the relevance of the terms relative to each other. This is useful, as it reduces the influence of web page length (which is assumed to be not related to the content of the web page) on web page similarity.

[2]Clustering algorithms which require a measure of *distance* can apply the *cosine distance*, which is defined as $1 - \varphi_{\cos}$.

## 2.1.2 Web Page Clustering

Finally, a clustering algorithm forms groups of similar web pages. Clustering algorithms employ the similarity between the represented web pages to group the web pages into a set of clusters $\mathcal{C}$. A cluster consists of web pages which are similar to each other, but dissimilar to web pages in other clusters. It is then assumed that the web pages in each cluster (and only these) concern the same referent. This *person clustering hypothesis* was explored by Balog et al. [2008]. As they found, even standard clustering algorithms can achieve surprisingly good results in this task.

Next, two well-known clustering algorithms are described: the *Single Pass Clusterer* (SPC) and the *Hierarchical Agglomerative Clusterer* (HAC). Both clustering algorithms form disjoint clusters without discarding any web page.

**Single Pass Clustering** Single pass clustering, as it is described by Balog et al. [2008], considers each web page only once (i.e., it makes only a single pass over the list). It starts with only one cluster containing the first web page. After that, the algorithm loops, selecting one of the remaining web pages in each iteration. Then, the cluster closest to the selected web page is identified as follows: if the similarity between the cluster and the web page is greater or equal to a threshold $\theta_{\text{SPC}}$, the web page is assigned to this cluster; otherwise, a new cluster which contains only the web page is created. The algorithm terminates if each web page was assigned to a cluster, or a pre-defined number of clusters is reached. The second termination condition is, however, not considered in this thesis, since it is reported to have a counterproductive effect in the context of web people search [Balog et al., 2008].

In order to compute the similarity between a web page and a cluster, the similarity between the web page and the centroid $\mathbf{c}$ of the cluster can be used. The centroid $\mathbf{c}$ of a cluster $C$ is defined as the vector most similar to all represented web pages in $C$.[3] If the cosine similarity is applied, the centroid can be calculated by:

$$\mathbf{c} = \sum_{d \in C} \frac{\mathbf{d}}{\|\mathbf{d}\|}$$

Then, the cluster centroids are used for the similarity computation:

$$\varphi_{\text{centroid}}(d, C) = \varphi_{\cos}(\mathbf{d}, \mathbf{c}) \tag{2.3}$$

The single pass clustering procedure is shown in Algorithm 2.1.

The final clustering of the single pass clusterer depends on the order of web pages. To estimate the effect of web page order on clustering quality, we

[3] The vector for which $\sum_{d \in C} \varphi(\mathbf{d}, \mathbf{c})$ is maximal.

**Algorithm 2.1:** Single Pass Clustering

```
Input: Set of web pages $D = \{d_1, \ldots, d_n\}$
Input: Similarity threshold $0 \leqslant \theta_{\text{SPC}} \leqslant 1$
Output: Set of clusters $\mathcal{C} = \{C_1, \ldots, C_m\}$
$\mathcal{C} \leftarrow \{\ \}$
foreach $d \in D$ do
    if $\mathcal{C} = \{\ \}$ then
        $\mathcal{C} \leftarrow \{\{d\}\}$                                /* initial cluster */
    else
        $C \leftarrow \arg\max_{C \in \mathcal{C}} \varphi_{\text{centroid}}(d, C)$
        if $\varphi_{centroid}(d, C) \geqslant \theta_{SPC}$ then
            $\mathcal{C} \leftarrow (\mathcal{C} \backslash \{C\}) \cup \{C \cup \{d\}\}$       /* merge with $C$ */
        else
            $\mathcal{C} \leftarrow \mathcal{C} \cup \{\{d\}\}$                       /* new cluster */
        end
    end
end
```

performed clustering experiments with random order. The effect was deemed negligible. Therefore, we maintain the order of the web pages as they appear in the search results as suggested by Balog et al. [2008].

**Hierarchical Agglomerative Clustering** Of the several clustering algorithms used for the web people search task, hierarchical clustering is the most common [Artiles et al., 2009]. The hierarchical agglomerative algorithm starts with each web page being assigned to a cluster containing only this web page (*singleton clusters*). In each iteration, the two most similar clusters are identified and then merged. The algorithm terminates as soon as a stopping criterion is met.

While the web page similarity is applied for clusters containing only one web page, there exist various strategies to calculate the similarity of clusters containing more than one web page. One of these strategies is *single link* (also called *nearest-neighbour* [Lance and Williams, 1967]), which is often employed in web people search. It is reported to work well [Chen et al., 2009, Nuray-Turan et al., 2009]. Other strategies are described in Lance and Williams [1967]. In the single link strategy, the similarity between two clusters is the maximum similarity between any web page of one and any web page of the other cluster. By utilizing the cosine similarity $\varphi_{\cos}$ (cf. Equation 2.2), the

**Algorithm 2.2:** Single Link Hierarchical Agglomerative Clustering

```
Input: Set of web pages $D = \{d_1, \ldots, d_n\}$
Input: Similarity threshold $0 \leqslant \theta_{\text{HAC}} \leqslant 1$
Output: Set of clusters $\mathcal{C} = \{C_1, \ldots, C_m\}$
$\mathcal{C} \leftarrow \{\{d\} : d \in D\}$
$continue \leftarrow$ TRUE
while $continue \wedge |\mathcal{C}| > 1$ do
    $\{C, C'\} \leftarrow \operatorname*{arg\,max}_{\{C,C'\}:C\in\mathcal{C} \wedge C'\in\mathcal{C}\backslash\{C\}} \varphi(C, C')$
    if $\varphi(C, C') \geqslant \theta_{HAC}$ then
        $\mathcal{C} \leftarrow (\mathcal{C}\backslash\{C, C'\}) \cup \{C \cup C'\}$
    else $continue \leftarrow$ FALSE
end
```

similarity between clusters is calculated as:

$$\varphi_{\text{single-link}}(C, C') = \max_{d\in C, d'\in C'} \varphi_{\cos}(\mathbf{d}, \mathbf{d}')$$

As a stopping criterion, a global similarity threshold can be employed.[4] The algorithm stops merging if the similarities between all clusters are below a predefined similarity threshold $\theta_{\text{HAC}}$. More formally, the algorithm terminates as soon as $\varphi_{\text{single-link}}(C, C') < \theta_{\text{HAC}}$ for all distinct clusters $C, C'$ in $\mathcal{C}$. Therefore, the threshold should be selected with respect to the applied similarity measure. The clustering procedure is shown in Algorithm 2.2.

## 2.1.3 Special Web Pages in Web People Search

Special web pages are web pages for which the assumption that each web page concerns exactly one referent with the target name does not hold. Two cases can be distinguished: multiple-referent pages and noise pages.

*Multiple-referent pages* refer to several individuals with the target name (e.g., "John F. Kennedy" and his son, "John F. Kennedy Jr."). In the final clustering, such web pages should ideally be part of multiple clusters. If a clustering algorithm is employed which is—like the algorithms introduced above—not able to assign web pages to multiple clusters, the page has to be segmented. The text passages which concern different referents are then treated as separate web pages in clustering. After the clustering, the original web page is assigned to each cluster at least one of the text passages was assigned to.

[4] A theoretical discussion of various stopping criteria can be found in Kleinberg [2002].

The second kind of special pages, which should not be placed in any cluster, are called *noise pages*. These web pages may only mention the target name, but not provide enough information to be assigned to any referent. Also, the target name may not exclusively refer to persons, but also to institutions, products or locations (e.g., “John Kennedy Steakhouse”). Once identified, noise pages are discarded from the clustering process.

### 2.1.4 Evaluation

For the evaluation of web page clustering, we apply the F-Measure of extended BCubed precision and recall [Amigó et al., 2009]. In comparison to other metrics for the evaluation of clustering, it is the only measure which satisfies the following four formal constraints: (1) cluster homogeneity, (2) cluster completeness, (3) rag bag and (4) cluster size vs. quantity [Amigó et al., 2009]. The employment of the extended BCubed evaluation metrics for the workshop is justified by Amigó et al. [2009].

The BCubed evaluation measures (precision and recall) are defined on a per-web-page base. Bagga and Baldwin [1998] defined BCubed document precision ($\mathrm{p}_{\mathrm{B}^3}$) and document recall ($\mathrm{r}_{\mathrm{B}^3}$) for each single web page in the set of clusters $\mathcal{C}$:

$$\begin{aligned}\mathrm{p}_{\mathrm{B}^3}(d) &= \frac{|\{d' \in D : (C(d') = C(d)) \wedge (r(d') = r(d))\}|}{|\{d' \in D : (C(d') = C(d))\}|} \\ \mathrm{r}_{\mathrm{B}^3}(d) &= \frac{|\{d' \in D : (C(d') = C(d)) \wedge (r(d') = r(d))\}|}{|\{d' \in D : (r(d') = r(d))\}|}\end{aligned} \tag{2.4}$$

with $C(d)$ denoting the cluster $d$ was assigned to and $r(d)$ being the referent the web page concerns. An example calculation of the two measures is given in Figure 2.3. The final BCubed precision ($\mathrm{P}_{\mathrm{B}^3}$) and recall ($\mathrm{R}_{\mathrm{B}^3}$) are then calculated by averaging the per-document values:

$$\mathrm{P}_{\mathrm{B}^3} = \frac{1}{|D|} \cdot \sum_{d \in D} \mathrm{p}_{\mathrm{B}^3}(d) \qquad \mathrm{R}_{\mathrm{B}^3} = \frac{1}{|D|} \cdot \sum_{d \in D} \mathrm{r}_{\mathrm{B}^3}(d)$$

These original BCubed measures are extended by Amigó et al. [2009] to allow for overlapping clusters. This is necessary if multiple-referent pages, which should be placed in multiple (therefore overlapping) clusters, are considered. If clusters do not overlap and no multiple-referent pages are present, the extended measures provide the same results as the metrics proposed by Bagga and Baldwin. The extended metrics are detailed in Appendix A on page 83.

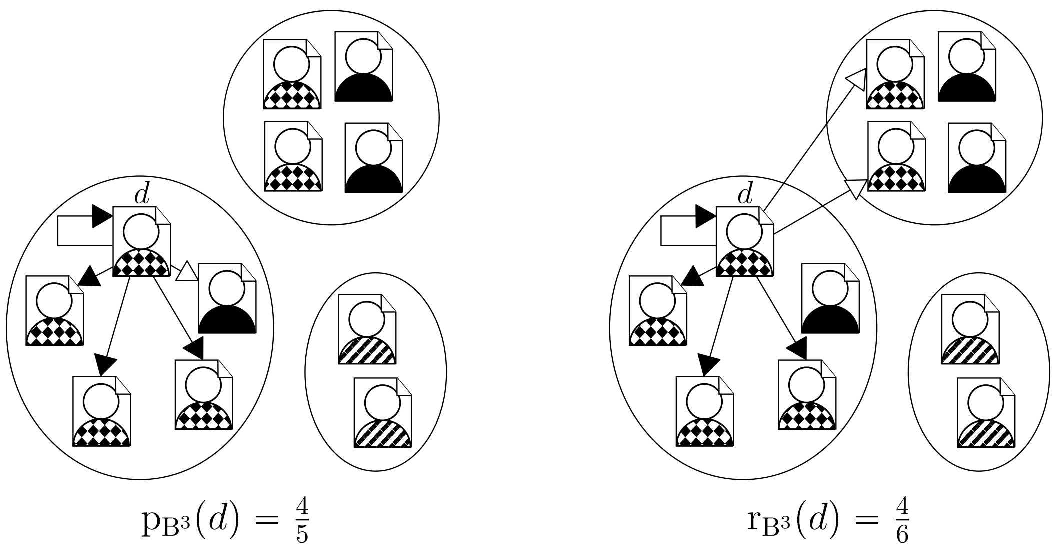


**Figure 2.3:** Example calculation of BCubed document precision (left) and recall (right) for a web page $d$. All web pages in the same ellipse are assigned to the same cluster. Decisions of correctly grouping a web page with $d$ are denoted by a black-headed arrow. White-headed arrows represent incorrectly grouped (left) or incorrectly separated (right) pages. This example is adapted from Amigó et al. [2009].

### F-Measure

In order to combine the two BCubed metrics into a single metric, we employ the F-Measure [Manning et al., 2008]. This measure weighs out gains in precision and recall by a parameter $\alpha \in [0, 1]$. This parameter has to be adjusted for the task: if precision is more important than recall, a value greater than 0.5 is used.

The F-Measure is derived from the effectiveness measure discussed in Rijsbergen [1974] and can be written as:

$$F_{\alpha=a} = \frac{1}{\frac{a}{\mathrm{P}} + \frac{1-a}{\mathrm{R}}} \tag{2.5}$$

The resulting score is the harmonic mean of precision and recall if $\alpha$ is 0.5.

It is argued by Artiles et al., that an increase in recall is preferred over an increase in precision for the task of web people search. It is assumed by the authors, that it is easier for a user to discard unwanted web pages, than to search through all other clusters for missing pages. Nevertheless, $F_{\alpha=0.5}$ was employed as the evaluation criterion in the workshop. It should also be noted that the overall F-Measure is calculated by averaging the F-Measure results of the single target names (*macro-average*).

## 2.2 Person Attributes

Previously, clustering algorithms were introduced to automatically group documents by the referents they concern. If the resulting clusters are presented to the user as they are, the user still has to identify the cluster which refers to the person of interest. In order to assist the user in this task, techniques of cluster labeling can be applied. If the labels are chosen appropriately, the user is able to find the desired cluster by only inspecting the usually very short labels. As the clusters refer to different referents, it is intuitive to employ personal characteristics of these individuals as cluster labels.

A person attribute is an abstraction from characteristics of multiple individuals (e.g., occupation as abstraction from actor, artist, writer, etc.). Sekine and Artiles [2009] created a list of attributes from web pages. They asked annotators to pick out all characteristics that could be expressed in the following form: person's *attribute* is *attribute-value*. By this, they identified 123 distinct person attributes. Since not all person attributes are equally helpful in web people search, Sekine and Artiles chose 16 of the 123 person attributes for the attribute extraction task of the web people search workshop (WePS-2).[5] They selected person attributes which are "general enough to cover most people, useful for disambiguation, and meaningful for the evaluation" [Sekine and Artiles, 2009]. The person attributes include

- Contact information such as email, phone, fax and an authorized website.
- The person's background such as the birthplace, date of birth and nationality. But also the names of relatives, mentors and attended schools, achieved degrees or a major.
- The line of work such as occupation and affiliation.
- Other information such as awards and other names (i.e., nicknames, pseudonyms and third names).

Since we want to compare person attributes in detail, we focus on 8 of these person attributes, which are described in Table 2.1.[6] The selected person attributes occur reasonable often and cover the different types of information.

[5]The WePS-2 corpus also features the person attribute values for the person attributes work and location. These two person attributes were, however, removed from the evaluation as their number of occurrences varies strongly between the different entities [Sekine and Artiles, 2009].

[6]The descriptions were adapted from the WePS-2 workshop definitions of the attribute extraction task: `http://nlp.uned.es/weps/weps-2/WePS2_Attribute_Extraction.pdf`

**Table 2.1:** The person attributes used in this thesis.

| Attribute | Description |
|---|---|
| Affiliation | An organization or group the referent is associated with. |
| Birthplace | The location (country, city, etc.) the referent was born at. We replace the U.S. state abbreviations with the full names and normalize country names (e.g., "United Kingdom", "Britain" and "UK"). |
| Date of birth | The date when the referent was born. |
| Email | A complete email address of the target person. |
| Nationality | A country name or an adjective of nationality for where the referent has citizenship. As for *birthplace*, country names are normalized. |
| Occupation | An occupation of the referent. |
| School | An institution (including a kindergarten, elementary school, middle school and high school) which the referent attended. |
| Third name | This is not a WePS-2 attribute, but uses some values of the official *other name* attribute. Other than using any name of the referent (including nicknames), values of this attribute are only those person names which appear together with the person's first and last name (e.g., John *Alexander* Kennedy). |

In the next section, a short introduction to current state-of-the-art techniques of attribute extraction is given. After that, it is discussed how inferences about person disambiguation can be drawn from extracted person attribute values. To judge the suitability of different person attributes for disambiguation, we propose a person attribute categorization based on the attribute-referent cardinality in Section 2.2.2.

### 2.2.1 Attribute Extraction

Most current algorithms for attribute extraction first identify candidates for person attribute values, and then decide whether a candidate value is indeed a person attribute value. Additionally, an attribute extraction system has to group the person attribute values by the individuals they belong to. The person attribute values which do not belong to a referent are then discarded. After the extraction, the person attribute values of each cluster are merged into consistent lists (cf. Figure 2.4).

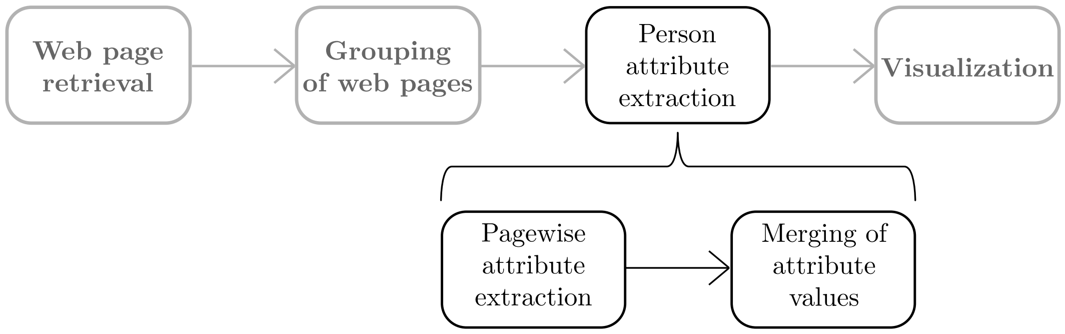

Web page retrieval
Grouping of web pages
Person attribute extraction
Visualization
Pagewise attribute extraction
Merging of attribute values

**Figure 2.4:** Attribute extraction as part of the web people search process.

### The PolyUHK System for Per-page Extraction

To give the reader an idea of how person attribute values can be extracted from web pages, the approach by Chen et al. [2009] is discussed as an example. The attribute extraction algorithm performed best in the WePS-2 workshop (cf. Section 4.1.1 on page 50).

The algorithm segments the web pages into single *fragments*. A fragment consists of all continuous lines of text which have the same writing style. Complete sentences indicate a so-called *formal* style. In contrast, the *informal* style is characterized by lists and line breaks which are used to present personal information. If the web page was detected to be a homepage of the referent by analyzing the web page title, person attribute values will be extracted from all fragments. If not, only those fragments which contain the target name are considered.

The person attribute values are then extracted using either keywords or manually created patterns. Some person attributes, such as email, are extracted from the fragments by applying regular expressions. Which patterns are applied depends on the style of the fragment. For example, if one line of informal text contains only the target name, it is assumed that the following lines contain further personal information: the next line is assumed to contain the occupation and the second next line the affiliation of the referent. Another method for extracting person attribute values is the use of keywords (e.g., “born” for date of birth and birthplace). In this case, named entities, such as locations and organisations, are discovered as candidates for person attribute values. In conjunction with a keyword, the person attribute of the candidate is detected. Since not all named entities are person attribute values, the named entity is discarded if no keyword is present in the same sentence.

In the evaluation of the WePS-2 workshop, the described method detected 8% of the person attribute values which were detected by human annotators. Furthermore, 30% of the extracted person attribute values correctly referred to person attributes of the referent [Sekine and Artiles, 2009]. As can be seen from these values, current algorithms for attribute extraction are far from perfect. Only every third extracted value was also found in the manual extraction. Moreover, more than 90% of the manually extracted person attribute values were missed.

#### Attribute Value Merging

If multiple person attribute values of one person attribute are extracted from the web pages of one cluster, it might be necessary to choose one or a few of them for the cluster visualization (cf. Figure 2.1 on page 4). A complete list of these values may be too long, or contain conflicting values. For example, a web page might contain an incorrect date of birth. Displaying two different dates is certainly not helpful for the user. On the other hand, inspecting the extracted values on a per-cluster basis might be able to account for errors in web pages as well as during the extraction. Additionally, value merging assists the user if variant forms of spelling of names—for example of organisation names—are merged. To the best of our knowledge, there is not much research about attribute value merging, although it was required in the third workshop on web people search, WePS-3 [Artiles et al., 2010]. Nagy and Farkas [2010], the winners of the WePS-3 attribute extraction task, simply select the most frequent attribute value.

### 2.2.2 Person Disambiguation via Attributes

Person attributes were introduced into web people search to assist the user in identifying the referents in the visualization of the web page clustering. To this point, it is only discussed that a user profits from person attributes when disambiguating the final clusters. Nevertheless, it is intuitive to use the extracted information as much as possible. Thus, related work is concerned with incorporating person attribute information into the clustering process: (1) by forming seed-clusters from identical person attribute values [Mann, 2006]; (2) by modifying the similarities between the web pages [Nagy and Farkas, 2010, Wan et al., 2005]; (3) or by creating a completely new similarity metric [Jiang et al., 2010]. The authors argue that if two web pages contain equal values for a person attribute, it is a clue that these two web pages concern the same referent. Moreover, we want to add that different values can imply that two web pages concern different referents.

Considering the selected attributes in Table 2.1 (cf. page 14), it can be seen that not all person attributes can provide the same information for disambiguation. Reconsider the example of the introduction: If two web pages concern a footballer "John Kennedy", then the probability that the web pages concern the same referent is higher than without this information. Nevertheless, it is still possible that there are multiple footballers with the same name. Different dates of birth, however, are, under the assumption that the two values are correct, a clear signal that the two pages concern different referents. Thus, not only the implication deduced from the attributes can be different (here grouping the footballers and separating different dates of births), but also the strength of the implication.

Next, we propose a person attribute categorization which corresponds to this difference in significance. After that, the proposed theoretical categorization is associated with statistical properties of attributes: the *referents-per-value* and *values-per-referent* ratio. Through this, it is possible to create a less strict categorisation, which can be more applicable in clustering. Our analysis of person attributes finally forms the foundation for the application of person attribute information in web page clustering in Chapter 3.

### Cardinality-based Attribute Categorisation

We propose to capture the intuition of different disambiguation capabilities by the term of cardinality as it is known in database design. The cardinality describes the possible number of relations between records in one table and records in a second table. Since each person has exactly one date of birth, but there are several persons born at this date, the date of birth attribute is a relation of type *one-to-many* (1:$n$).[7] Thus, different values in these attributes clearly indicate different referents. The email address is of another type: although a person may have multiple email addresses, only one particular person is associated with each of them. It is therefore a relation of type *many-to-one* ($n$:1).

Most of the attributes selected by Sekine and Artiles, however, are of type *many-to-many* ($n$:$m$). Thus, referents can have several values for these person attributes associated with them, and a value for these person attributes can be associated with several referents. Although grouping decisions can not be deduced from these person attributes, the values for these person attributes can still be seen as clues (cf. the footballer example). The last type is *one-to-one* (1:1); it contains person attributes for which each value is associated with exactly one referent and for which each referent has only this value. A hypothetical global Social Security number would be an example of this attribute type.

Using this observation, all person attributes can be categorised into four different groups, which represent the four possible *cardinalities*. An example

[7] We arbitrarily decided for this order, but will use it consistently.

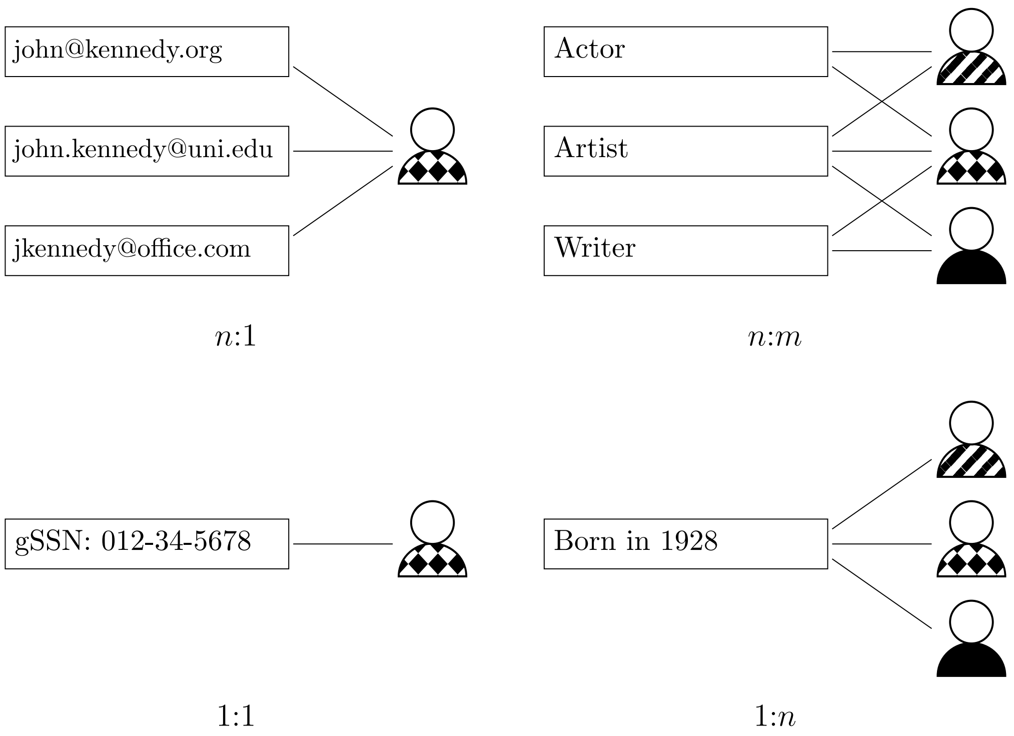


**Figure 2.5:** The four different categories of person attributes; each category is represented by an example: email address ($n$:1), occupation ($n$:$m$), a hypothetical global Social Security number (1:1) and date of birth (1:$n$).

of each group is shown in Figure 2.5.

### Referents-per-value and Values-per-referent Ratio

Although this categorisation is a theoretical foundation for the employment of person attributes for disambiguation, it should be extended to a continuous scale for practical considerations. Person attributes with a *one-to-x* cardinality suggest a separation of web pages if the values for the person attribute mismatch. On the other hand, equal values for person attributes with a cardinality of *x-to-one* are a reason for grouping web pages. Nevertheless, we do not deem *many-to-many* person attributes useless for disambiguation. Some person attributes have, for example, only few—but possibly more than one—values for one referent (e.g., in the case of the nationality person attribute). We thus extend the categorization to a continuous scale by introducing the *referents-per-value* (rpv) and *values-per-referent* ratio (vpr) of an person attribute $a$:

$$\begin{aligned}\text{rpv}(a) &= \frac{1}{|V_a|} \cdot \sum_{v \in V_a} |\{r : r \in R \wedge v \in V_{a,r}\}| \\ \text{vpr}(a) &= \frac{1}{|\{r : r \in R \wedge V_{a,r} \neq \{\,\}\}|} \cdot \sum_{r : r \in R \wedge V_{a,r} \neq \{\,\}} |V_{a,r}|\end{aligned} \tag{2.6}$$

with $V_a$ being the set of known values for the person attribute $a$ and $V_{a,r}$ denoting all person attribute values for $a$ which are connected to referent $r$. We assume that the clue for grouping together two web pages because of a shared person attribute value becomes stronger with an decreasing values-per-referent ratio of the person attribute. Similarly, clues for the separation of web pages become stronger with an decreasing referents-per-value ratio. Both ratios have a lower bound of 1 and an upper bound of the number of values which exist for the person attribute (values-per-referent ratio) and the maximum number of individuals who share a name (referents-per-value ratio) respectively. It should be noted, that these ratios are statistical values and require the correctly extracted person attribute values to be calculated. The cardinality, on the other hand, is logically deduced from knowledge about the person attributes.

In order to relate the cardinality to the values-per-referent and referents-per-value ratios, a collection of annotated web pages (the WePS-2 corpus) was analyzed. This collection features the result pages of 30 person name queries. The annotation of the web pages includes the person attributes values which could be extracted from each web page as well as the assignment of each web page to a referent. More information on the collection is provided in Section 4.1.1 on page 50. For each of the person attributes in Table 2.1 on page 14, the values-per-referent and referents-per-value ratios were calculated using the annotation.[8] We calculated both ratios for each of the 30 sets of result pages separately and computed the average, the minimum and the maximum ratio for the 30 results. Additionally, we calculated both ratios for the joint set of all result pages. The results, together with the cardinality, are provided in Table 2.2.

Four points should be noted:

1. Relatively many values are found per referent, even for attributes which are theoretically bounded by one value per referent. Reasons for this are factual errors in the web pages and insufficient value-matching algorithms, especially for birthplace. The different person attribute values for

[8]The calculation also requires a method of recognizing identical attribute values. For example, the dates "November 18th, 1978" and "1978/11/18" should be treated as the same value. To produce results which can be associated with our experiments, the same methods, which are described in Section 3.2.1 on page 29, are applied.

**Table 2.2:** Cardinality (c) and values-per-referent and referents-per-value ratios of the selected person attributes in the WePS-2 corpus: (1) the average ($\mu$), minimum ($\underline{x}$) and maximum ($\overline{x}$) of the calculated ratios for each target name $tn$; (2) the calculated ratios for all target names in $TN$ at once.

| | | values-per-referent | | | | referents-per-value | | | |
|---|---|---|---|---|---|---|---|---|---|
| Attribute | c | $\mu_{\text{vpr}_{tn}}$ | $\underline{\text{vpr}_{tn}}$ | $\overline{\text{vpr}_{tn}}$ | $\text{vpr}_{TN}$ | $\mu_{\text{rpv}_{tn}}$ | $\underline{\text{rpv}_{tn}}$ | $\overline{\text{rpv}_{tn}}$ | $\text{rpv}_{TN}$ |
| Affiliation | $n$:$m$ | 8.36 | 2.00 | 37.00 | 4.92 | 1.00 | 1.00 | 1.02 | 1.03 |
| Birthplace | 1:$n$ | 2.44 | 1.00 | 12.00 | 1.69 | 1.00 | 1.00 | 1.00 | 1.05 |
| Date of birth | 1:$n$ | 1.11 | 1.00 | 2.00 | 1.12 | 1.05 | 1.00 | 2.00 | 2.22 |
| Email | $n$:1 | 1.30 | 1.00 | 3.00 | 1.29 | 1.01 | 1.00 | 1.14 | 1.01 |
| Nationality | $n$:$m$ | 1.18 | 1.00 | 2.00 | 1.30 | 1.00 | 1.00 | 1.00 | 1.30 |
| Occupation | $n$:$m$ | 9.60 | 1.50 | 55.00 | 4.42 | 1.04 | 1.00 | 1.18 | 1.44 |
| School | $n$:$m$ | 2.81 | 1.00 | 9.00 | 2.24 | 1.01 | 1.00 | 1.14 | 1.10 |
| Third name | $n$:$m$ | 1.02 | 1.00 | 1.44 | 1.06 | 1.07 | 1.00 | 1.33 | 1.87 |

birthplace are the names of the town, near city, region, state or country in which the person was born. We have not yet implemented a method that considers these different ways of expressing the same location. Because of the difference in the values-per-referent ratio, it is expected that mismatches in date of birth can lead to more accurate decisions for separating documents than mismatches in the birthplace.

2. The attributes nationality and third name, which are not theoretically bounded by one value per referent, have still a very low values-per-referent ratio. Therefore, we expect them to lead to similar results as date of birth when applied in disambiguation.

3. The maximum values of the values-per-referent ratios for the affiliation, the birthplace, the occupation and the school are considerably higher than the minimum values. Consequently, we assume that the application of these person attributes in web page clustering will also lead to different results for web page collections retrieved for different target names.

4. The average referents-per-value ratio is significantly low for each attribute. This is due to the relatively small amount (average of 85) of web pages per target name in the collection. If calculated on the whole corpus (~2800 documents, cf. $\text{vpr}_{Tn}/\text{rpv}_{Tn}$), the ratio is generally higher. Nevertheless, the small number of web pages is realistic if only the web pages retrieved by search results are considered, such as in this thesis.

The use of person attributes for person disambiguation as described above assumes that the extracted person attribute values are correct. However, a web people search system which deduces grouping decisions from person attributes should account for errors in the attribute extraction or in the web pages. It is thus important to note that decisions for grouping web pages are less sensitive to incorrect person attribute values than decisions for the separation of web pages. In the case of person attributes which are employed for grouping together web pages, a clue is missed because of the incorrect value. However, if the incorrect value is not associated with a different referent, no incorrect decision is derived from it. The situation changes when person attributes are considered as clues for the separation of web pages: if the same error is not repeated for/in the other web pages which concern the same referent, these web pages will be incorrectly separated.

In conclusion, we have shown that attribute values are suitable clues for web people search. Especially the low referents-per-value ratios suggest that referents can be identified by their attribute values. Still, this identification is heuristically. Therefore, it will also cause additional disambiguation errors. Nevertheless, we assume that the advantages outweigh the disadvantages. Moreover, by the employment of a combination of the person attributes, it might be possible to prevent the additional errors. Particularly, it seems promising to weigh out clues for the grouping and for the separation of documents. Although there are already some clustering methods that allow for the incorporation of attribute information (e.g., [Mann, 2006, Nagy and Farkas, 2010, Wan et al., 2005]), they only support information that suggests a grouping of web pages. Thus, we propose to use a different method for the integration of attribute information: *constrained clustering*. In the next chapter, this method is described and a framework for the application of constrained clustering in web people search is detailed. This framework is then evaluated in Chapter 4.

# Chapter 3

# Constraints in Web Page Clustering

We discuss in this chapter the integration of person attributes into web page clustering via *constrained clustering*. In order to allow for this integration, the steps in web people search (cf. Chapter 2, Figure 2.4 on page 15) have to be rearranged; the person attributes have to be extracted prior to clustering in order to employ them to guide the clustering process (cf. Figure 3.1). In order to incorporate the person attribute information into clustering, we employ constrained clustering algorithms, which perform semi-supervised learning (cf. Chapter 3.1). The utilization of person attributes in constrained clustering algorithms, however, requires the representation of the person attribute information in the form of constraints. We propose a framework for the generation of the constraints, which consists of person attribute extraction, person attribute value matching, constraint strength addition and transitivity/entailment computation (cf. the lower part of Figure 3.1). The first part of this framework is identical to the attribute extraction as introduced in Section 2.2.1. The

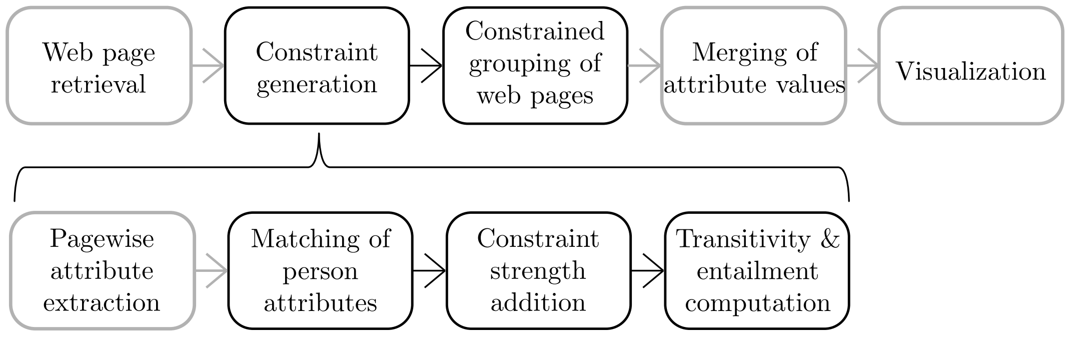

Web page retrieval
Constraint generation
Constrained grouping of web pages
Merging of attribute values
Visualization
Pagewise attribute extraction
Matching of person attributes
Constraint strength addition
Transitivity & entailment computation

**Figure 3.1:** Adjustment of the web people search process in order to use knowledge derived from person attributes in web page clustering.

remaining parts are discussed in Section 3.2. Finally, metrics for quantifying the quality of full sets of constraints in regard to their correctness and coverage are proposed in Section 3.3. These metrics are then used to evaluate our framework in Chapter 4.

## 3.1 Semi-supervised Learning

The clustering algorithms which were introduced in Section 2.1.2 on page 8 group web pages solely based on their textual content; by the choice of the web page representation, an assumption is made about which of the textual information is useful for referent disambiguation. Besides the choice of the parameters of the clustering algorithm, no further knowledge about the task domain, web people search, is provided to the clustering algorithm. Therefore, clustering algorithms are classified as algorithms for *unsupervised learning* [Manning et al., 2008]. Person attribute information, however, is domain-specific knowledge of the web people search task. For example, web pages which refer to persons with different dates of birth concern different referents. Therefore, it might be efficient to represent web pages by the person attributes values which are extracted from them. Then, rules for grouping the web pages based on person attribute values could be generated. If these rules are inferred from example web pages for which the concerned referent is known, the employed method is classified as a *supervised learning* algorithm [Manning et al., 2008]. An issue with this approach is that not all web pages contain person attributes. Supervised learning, however, requires that at least a single person attribute is contained in each web page. Hence, supervised learning which relies solely on person attributes is unsuitable for solving the problem of web people search.

Thus, a method is required which is able to exploit domain specific knowledge, but does not depend on it. Two methods are known for the incorporation of domain specific knowledge into the clustering process: (1) methods which change the similarity between a pair of documents if knowledge about their relation exists; (2) methods which directly enforce the grouping or separation of documents if knowledge about their relation exists. These methods employ domain specific knowledge such as in supervised learning, but apply unsupervised clustering in cases when no knowledge about the relation is present. Therefore, these methods are referred to as *semi-supervised learners* [Wagstaff, 2002]. We assume that person attributes are well-suited for referent disambiguation and knowledge deduced from person attributes should therefore be directly applied if existent. Therefore, we choose hard constrained clustering, which is a method of the second type, for the web people search task.

*Constrained clustering*, proposed by Wagstaff and Cardie [2000], is a well-known method of semi-supervised learning. Knowledge is modeled in the form of pairwise grouping decisions between two *instances*:[1] so-called *pairwise instance-based constraints*. The two types of constraints introduced by Wagstaff and Cardie suffice for the representation of the knowledge which can be deduced from person attributes:

- A *must-link* constraint $c_=$ between two web pages $d$ and $d'$, which we denote as $c_=(d, d')$, states that these two web pages have to be placed into the same cluster. Must-link constraints are employed if the person attribute values implies a grouping of two web pages (e.g., if two web pages concern people with the same email address).
- A *cannot-link* constraint $c_{\neq}$ between two web pages $d$ and $d'$, denoted as $c_{\neq}(d, d')$, states that these two web pages must *not* be placed into the same cluster. Cannot-link constraints are employed if the person attribute values implies a separation of web pages (e.g., if two web pages concern people with different dates of birth).

An introduction to the concepts of constrained clustering is given in Section 3.1.1. Although this section is useful for a better discernment of constrained clustering, it is not necessary for the understanding of the remaining parts of this thesis. After that, the two clustering algorithms which are described in Section 2.1.2 are extended to enforce instance-based constraints in the clustering process. The representation of knowledge inferred from person attributes in the form of must-link and cannot-link constraints is then detailed in Section 3.2.

### 3.1.1 Excursion: Concepts of Constrained Machine Learning

Domain knowledge has been employed in clustering for a long time. An early example is the clustering of layers along a drill hole by Gordon [1973]: all layers within a cluster had to be adjacent. From a more theoretical perspective, Mitchell [1980] describes factual knowledge as one of several classes of *biases* in machine learning. Machine learning algorithms have to choose one solution of the set of all possible solutions. If the algorithm does not pick the solution randomly, it is biased towards solutions of specific kind (e.g., more "simple" solutions). Gordon and Desjardins [1995] distinguish between representational

[1] Instances is the general term for the single items that are grouped or classified in machine learning. In the case of web page clustering, the web pages are the instances.

and procedural bias: representational bias limits the possible states of an algorithm; procedural bias has an impact on which of the possible states the algorithm chooses.

Instance-based constraints are therefore a form of representational bias, as they prohibit all states in which they are violated (i.e., states in which instances connected by must-link constraints are separated or cannot-linked instances are in the same cluster). Nevertheless, instance-based constraints can also form a procedural bias: (1) *soft constraints* are not imposed on the clustering algorithm, but only suggest a preference for choosing a state [Wagstaff, 2002]; (2) *metric learning algorithms* try to infer generalizations from the constraints and apply them to other instances [Klein et al., 2002].

One advantage of instance-based constraints is that they are a comprehensible form of domain knowledge. Therefore, they can be imposed on a clustering process by the user if clearly incorrect decisions are spotted. Then, the clustering process is rerun, enforcing the correction. Similarly, an algorithm can identify crucial decisions and then ask the user to decide if to group or separate the instances (*active learning*) [Basu et al., 2004]. The decision of the user can then be represented as a constraint. Furthermore, users can also apply instance-based constraints to test hypothesises about the data [Talavera and Béjar, 1999].

### 3.1.2 Constrained Clustering Algorithms

Most clustering algorithms can be easily modified to enforce instance-based constraints. Wagstaff [2002] describes a specific procedure to modify partitioning clustering algorithms. Only a few modifications are required in the case of the algorithms described in Section 2.1.2 on page 8.

In order to enforce constraints, clusters are created at the start of an algorithm such that all web pages which are connected by a must-link are in the same cluster. The function *apply-must-links*$(D, Con_=)$ (see there) is employed to create these clusters. It should be noted that this function considers the *transitivity* of must-link constraints. Therefore, if two must-link constraints $c_=(d, d')$ and $c_=(d', d'')$ exist, also $d$ and $d''$ will be in the same cluster as if they were linked by a further must-link constraint. It should further be noted that this might result in conflicting constraints when $d$ and $d''$ are also linked by a cannot-link constraint. We will discuss how to resolve such conflicts in Section 3.2.3 on page 40 and assume for the rest of this section that no such conflicts exist.

Some clustering algorithms split clusters after their initialization in order to optimize the clustering. In this case, the algorithm has to be further modified to prevent the separation of web pages which were grouped because of must-

```
Function apply-must-links($D$, $Con_=$)
    Input: Set of web pages $D = \{d_1, \ldots, d_n\}$
    Input: Set of must-links $Con_= = \{c_{=,1}, \ldots, c_{=,p}\}$
    Output: Set of clusters $\mathcal{C} = \{C_1, \ldots, C_m\}$
    $\mathcal{C} \leftarrow \{\{d\} : d \in D\}$
    foreach $c_=(d, d') \in Con_=$ do
        $C \leftarrow C : C \in \mathcal{C} \wedge d \in C$            /* get cluster of $d$ */
        $C' \leftarrow C : C \in \mathcal{C} \wedge d' \in C$          /* get cluster of $d'$ */
        if $C \neq C'$ then
            $\mathcal{C} \leftarrow (\mathcal{C} \backslash \{C, C'\}) \cup \{C \cup C'\}$     /* merge $C$ with $C'$ */
        end
    end
```

link constraints. Since neither the single pass clusterer nor the hierarchical agglomerative clusterer split clusters after they were formed, we do not further discuss this issue. The interested reader is referred to Section 3.1.2 of [Wagstaff, 2002].

For both of the introduced clustering algorithms, cannot-links have only to be considered when the candidate clusters for merging two clusters are calculated. We therefore define *get-valid-merges-for-cluster*$(C, \mathcal{C}, Con_{\neq})$, which is the function that returns all clusters of the clustering $\mathcal{C}$ the cluster $C$ can be merged with under consideration of the set of cannot-link constraints $Con_{\neq}$. (Therefore, all clusters in $\mathcal{C}$ which contain no web page which is linked to a web page contained in $C$ via a cannot-link constraint.) Although cannot-links constraints are not transitive, and therefore $c_{\neq}(d, d'')$ can not be deduced from $c_{\neq}(d, d')$ and $c_{\neq}(d', d'')$, cannot-link constraints fulfill the *entailment* property in conjunction with must-link constraints [Davidson et al., 2006]: if there exist a cannot-link constraint $c_{\neq}(d, d')$ and a *must-link* constraint $c_=(d', d'')$, then the cannot-link constraint $c_{\neq}(d, d'')$ can be logically deduced from this situation.

## Constrained Single Pass Clustering

There are multiple ways of extending the single pass clusterer, which is described in Algorithm 2.1 on page 9, for instance-based constraints; *must-link* constraints can either be considered prior to the actual algorithm (such as by using the function *apply-must-links*), or each time a web page is included into the clustering. *Cannot-link* constraints are in both cases considered each time a web page is included into the clustering. We employ the first method in our experiments since it further reduces the dependency of the clustering algorithm on the order in which the web pages are considered. This method,

**Algorithm 3.1:** Constrained Single Pass Clustering

```
Input: Set of web pages $D = \{d_1, \ldots, d_n\}$
Input: Set of must-links $Con_= = \{c_{=,1}, \ldots, c_{=,p}\}$
Input: Set of cannot-links $Con_{\neq} = \{c_{\neq,1}, \ldots, c_{\neq,q}\}$
Input: Similarity threshold $0 \leqslant \theta_{\mathrm{SPC}} \leqslant 1$
Output: Set of clusters $\mathcal{C} = \{C_1, \ldots, C_m\}$
$\mathcal{C} \leftarrow \{\,\}$
$\mathcal{C}' \leftarrow$ apply-must-links$(D, Con_=)$
foreach $C' \in \mathcal{C}'$ do
    $\mathcal{C}'' \leftarrow$ get-valid-merges-for-cluster$(C', \mathcal{C}, Con_{\neq})$
    if $\mathcal{C}'' = \{\,\}$ then
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{C'\}$                    /* create new cluster */
    else
        $C \leftarrow \arg\max_{C \in \mathcal{C}''} \varphi_{\mathrm{centroid}'}(C', C)$
        if $\varphi_{centroid'}(C', C) \geqslant \theta_{SPC}$ then
            $\mathcal{C} \leftarrow (\mathcal{C} \backslash \{C\}) \cup \{C \cup C'\}$          /* merge $C'$ with $C$ */
        else
            $\mathcal{C} \leftarrow \mathcal{C} \cup \{C'\}$                /* create new cluster */
        end
    end
end
```

however, requires a similarity metric for the comparison of two clusters, since web pages can already be grouped before they are considered during the main part of the algorithm. Therefore, the centroid similarity, which is defined in Equation 2.3 on page 8, is adapted for the comparison of two clusters:

$$\varphi_{\mathrm{centroid}'}(C_i, C_j) = \varphi_{\cos}(\mathbf{c}_i, \mathbf{c}_j)$$

Constrained single pass clustering is defined in Algorithm 3.1.

### Constrained Hierarchical Agglomerative Clustering

The incorporation of constraints into the hierarchical clustering procedure (cf. Algorithm 2.2 on page 10) is very similar to the previously detailed incorporation of constraints into the single pass clusterer. Even more, the enforcement of must-link constraints is identical to the merging of two clusters in the unconstrained clustering procedure. In order to incorporate cannot-link constraints, the function *get-valid-pairs-for-merging*$(\mathcal{C}, Con_{\neq})$ is applied, which returns all pairs of clusters in $\mathcal{C}$ that are not connected by any cannot-link constraint in

**Algorithm 3.2:** Constrained Single Link Hierarchical Clustering

```
Input: Set of web pages $D = \{d_1, \ldots, d_n\}$
Input: Set of must-links $Con_= = \{c_{=,1}, \ldots, c_{=,p}\}$
Input: Set of cannot-links $Con_{\neq} = \{c_{\neq,1}, \ldots, c_{\neq,q}\}$
Input: Similarity threshold $0 \leqslant \theta_{\text{HAC}} \leqslant 1$
Output: Set of clusters $\mathcal{C} = \{C_1, \ldots, C_m\}$
$\mathcal{C} \leftarrow \textit{apply-must-links}(D, Con_=)$
$\textit{continue} \leftarrow$ TRUE
while $\textit{continue} \wedge |\mathcal{C}| > 1$ do
    $V \leftarrow \textit{get-valid-pairs-for-merging}(\mathcal{C}, Con_{\neq})$
    if $V \neq \{\ \}$ then
        $\{C, C'\} \leftarrow \arg\max_{\{C,C'\} \in V} \varphi_{\text{single-link}}(C, C')$
        if $\varphi_{\textit{single-link}}(C, C') \geqslant \theta_{HAC}$ then
            $\mathcal{C} \leftarrow (\mathcal{C} \backslash \{C, C'\}) \cup \{C \cup C'\}$
        else $\textit{continue} \leftarrow$ FALSE              /* threshold reached */
    else $\textit{continue} \leftarrow$ FALSE                 /* no valid merges left */
end
```

the set of cannot-link constraints $Con_{\neq}$. This function is applied in the calculation of the most similar pair of clusters. It is used to filter out all pairs of clusters which can not be merged because of a constraint. The constrained single link hierarchical agglomerative clusterer is defined in Algorithm 3.2.

## 3.2 From Person Attributes to Constraints

In order to generate constraints for web page clustering from the person attributes which are extracted from the web pages, attribute values have to be matched; the information for each pair of web pages has to be represented by a single constraint; and the transitive closure of these decisions has to be taken. In the first step, it is checked for each pair of web pages if they share person attribute values (cf. Section 3.2.1). Second, all derived clues for the grouping and separation of pairs of web pages are collected, weighed out and represented by must-link and cannot-link constraints (cf. Section 3.2.2). The weighing is done in multiple steps; first all matches *within* each person attribute are considered; then the matches of all person attributes (*between* attributes). Finally, the transitivity and entailment properties of the constraints are applied (cf. Section 3.2.3). Conflicting constraints are resolved during this step.

### 3.2.1 Matching Person Attribute Values

In order to identify which web pages have the same and which have different values in a person attribute, a method of matching the person attribute values is required. Therefore, we define a function for matching two person attribute values $v, v'$ of one person attribute $a$; the function $match_a(v, v')$ returns the *matching score* of the two values. The score lies in the range from 0 (no match) to 1 (*exact match*). Scores between 0 and 1 signal a *partial match.* We consider partial matching to account for typographical errors (e.g, "Harvard" and "Havrad") and variant spellings (e.g., "New York City" and "New York"); although the person attribute values in the examples are clearly different, it is assumed that the same sense is meant. Therefore, $match_a$ should return a score near 1. Nevertheless, the person attribute values are clearly different, and in some cases single letters can change the meaning. Thus, the score should be less than 1. In order to test if it is beneficial to consider partial matches, we distinguish two methods in our experiments:

1. If *soft matching* is employed, the matching scores for the values of a person attribute $a$ are calculated using the matching function $match_a$.

2. If *exact matching* is employed, the matching scores are also calculated using $match_a$, but then rounded. During the generation of must-link constraints, the matching score will be 1 if two person attribute values are identical with respect to $match_a$ (i.e., $match_a = 1$) and 0 otherwise. If, however, the person attribute values are matched in order to generate a cannot-link constraints, the matching score will be 0 if two person attribute values are completely different with respect to $match_a$ (i.e., $match_a = 0$) and 1 otherwise. In both cases, constraints are then only generated in the extreme cases.

We will now continue by defining the different matching operators for the different person attributes.

For most of the person attributes we selected (cf. Table 2.1 on page 14), the same function $match_a$ is employed: for the person attributes affiliation, birthplace, email, nationality, occupation and school we match the person attribute values case-insensitive and by employing the soft-tf·idf string similarity metric [Cohen et al., 2003] in combination with the Jaro-Winkler metric [Winkler, 2006]. For the computation of the matching score, the soft-tf·idf metric compares the single words of the two person attribute values. (In the previous example, "New", "York" and "City" would each be compared with "New" and "York".) Therefore, it is able to detect variant spellings more easily than methods which consider the whole person attribute value at once. Moreover, soft-tf·idf is reported to work especially well in the task of named entity matching [Cohen et al., 2003]. Nevertheless, as noted by Moreau et al. [2008], the original soft-tf·idf metric is not symmetric and not bounded by 1, if one word of one person attribute value matches best with multiple values of the other person attribute. We therefore apply an adjusted version of the soft-tf·idf metric that circumvents these problems. The Jaro, Jaro-Winkler and soft-tf·idf metric including our adjustment are detailed in Appendix B on page 85.

Special matching functions are applied for both the third name and birth date person attributes. Often, only the initial or the first few letters of a third name are given. Thus, we also count it as a match if one person attribute value is a prefix of the other. If this is not the case, the soft-tf·idf metric is employed like in the cases listed above. Similarly, instead of a full date of birth, often only the birth year is given in a web page. We therefore separate the contained values for birth date into a day, month and year-part. Two values for date of birth match, if they are *semantically* identical[2] in at least one of the parts and are contradictory in none of them. Therefore, parts which are missing in one of the two person attribute values are not considered for the comparison (e.g., "4/25" matches with "April 1989"). We also perform partial matching for the date of birth person attribute in order to deal with factual errors in the web pages. If two values for a date of birth disagree only on the day, but agree on the month or year, the score of $match_{date\ of\ birth}$ is lowered to 0.75. Likewise, 0.50 is used if the dates agree only on the year, and 0.25 if they differ in the precise year, but not in the decade (e.g., both referent are born in the nineties). It should be noted that the values were chosen by hand and are not adjusted to the data.

At this stage, all scores are considered as clues for grouping web pages. In the remaining steps, the score $s$ between two web pages $d$ and $d'$ is used as the constraint strength of a soft must-link constraint $c_=(d, d', s)$ [Wagstaff, 2002]; if the strength is 0, no confidence is placed in the constraint; if the strength is 1, then there is no doubt that the constraint is correct. Since, however, even a exact match in person attribute values does not suffice to undoubtedly deduce that the generated constraint is correct, we scale the strength during the addition appropriately.

The result of this step, which is the collection of all such must-link constraints between each pair of web pages for a single attribute $a$, is denoted with $Con_a(d, d')$. It is calculated by:

$$Con_a(d, d') = \{c_=(d, d', match_a(v, v')) : v \in V_{a,d} \wedge v' \in V_{a,d'}\} \tag{3.1}$$

[2]For example, "april" and "apr" are semantically identical. But also "4" if evidence is present that it does not represent the day of the month.

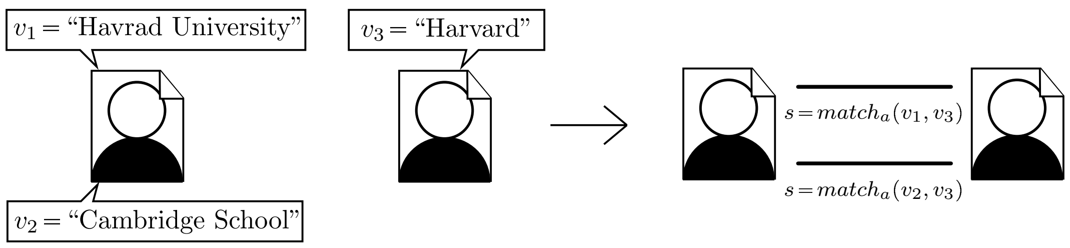


**Figure 3.2:** During person attribute value matching, each pair of person attribute values for each person attribute is matched for each pair of web pages. The resulting $c_=$ are represented by solid lines with the calculation of the strength $s$ shown below them.

with $V_{a,d}$ being all person attribute values of the person attribute $a$ which were extracted from web page $d$. It should be noted that a soft must-link constraint is also created if the two values do *not* match. These constraints have a strength of 0 and will be referred to as *neutral constraints* ($c_\sim$) in the remaining of this thesis. An example of this calculation is shown in Figure 3.2.

### 3.2.2 Constraint Strength Addition

In the next step, the constraint strengths of all soft constraints are added up to result in one soft constraint per pair of web pages. The conflicting decisions that may arise from taking the addition are then resolved in the last part (cf. Section 3.2.3). Constraint strengths close to the maximum of 1 within a set of constraints $Con_a(d, d')$ suggest that the web pages should be grouped together. The closer the scores are to 1, the stronger is the clue for grouping. On the other hand, scores close to 0 are a hint for the separation of two web pages if no score close to 1 is contained in $Con_a(d, d')$. Scores close to 0 together with a score close to 1 are instead a hint for the grouping of the web pages, since a referent might have multiple values for a single person attribute (cf. Table 2.2 on page 20). Thus, the low scores might also result from the comparison of different person attribute values of the same referent. Therefore, if the cardinality of the person attribute is not 1:$n$ as it is in the case of date of birth, two web pages are not separated because of evidence for the separation, but of lacking evidence for the grouping. Nevertheless, as will be shown in the experiments, this method still performs well. Moreover, if no scores exist—thus at least one of the web pages provides no values for the person attribute—no decision is derived from this situation. If this is the case for all considered person attributes, no constraint is generated to guide the clustering algorithm.

If soft constraints are generated for a pair of web pages, we proceed as follows:

1. The soft constraints for each person attribute are added up to a single score per pair of web pages, which is then interpreted as a clue for the grouping or the separation of the web pages.

2. The clues that originate from the different attributes are weighed out and a single soft constraint is generated from them.

Next, we the introduce the reader into constraint strength addition. Then, we further detail how constraint strength addition is employed to add up all soft constraints of a pair of web pages to a single one.

### Methods for Constraint Strength Addition

In order to combine the soft constraints into a single constraint, a method for constraint strength addition is required. To the best of our knowledge, however, no such method exists in the research literature. We now continue by introducing and discussing desired properties for such methods. Then, we propose two methods for constraint strength addition: *maximum* and *multiplication.*

**Desired Properties** In order to develop methods for soft constraint addition, we created a list of 9 desired properties of such methods. Similar to the notation of soft must-link constraints between two documents ($c_=(d, d', s)$), we denote soft cannot-link constraints by $c_{\neq}(d, d', s)$. Furthermore, a neutral constraint is denoted by $c_\sim(d, d', 0)$, with $c_\sim(d, d', 0) \equiv c_=(d, d', 0) \equiv c_{\neq}(d, d', 0)$.

**(1) Constraint strengthening** Adding a soft must-link constraint to a second soft must-link constraint must not result in a must-link constraint that is weaker than the strongest of them: $c_=(d, d', s) + c_=(d, d', s') = c_=(d, d', s'')$ with $s'' \geqslant \max(s, s')$. The same is true when cannot-link constraints are considered.

**(2) Constraint weakening** Adding up a soft must-link constraint and a soft cannot-link constraint must not result in a constraint which is stronger than any of them.

**(3) Neutral constraint** Adding a neutral constraint to a soft constraint should leave the second constraint unaffected:

$$
\begin{aligned}
c_=(d, d', s) + c_\sim(d, d', 0) &= c_=(d, d', s) \\
c_{\neq}(d, d', s) + c_\sim(d, d', 0) &= c_{\neq}(d, d', s) \\
c_\sim(d, d', 0) + c_\sim(d, d', 0) &= c_\sim(d, d', 0)
\end{aligned}
$$

**(4) Inverse constraints** Adding a soft must-link constraint to a soft cannot-link constraint with equal strength must result in a neutral constraint:

$$c_{=}(d, d', s) + c_{\neq}(d, d', s) = c_{\sim}(d, d', 0)$$

**(5) Associative property and commutative law** The order in which soft constraints are added must not have any impact on the resulting constraint. Thus, for any constraints $c_a, c_b, c_c$ which connect the same web pages, the equations $c_a + c_b = c_b + c_a$ and $(c_a + c_b) + c_c = c_a + (c_b + c_c)$ hold.

**(6) Type equality** If the type, but not the strengths of the addends of any sum are exchanged, then also the resulting constraint will change the type, but not the strength. The type is not changed if the constraint is a neutral constraint. Thus, for example, if $s < s'$:

$$\begin{aligned} &c_{=}(d, d', s) + c_{\neq}(d, d', s') = c_{\neq}(d, d', s'') \\ \Leftrightarrow\ &c_{=}(d, d', s') + c_{\neq}(d, d', s) = c_{=}(d, d', s'') \end{aligned}$$

**(7) Closeness** Adding a constraint to a second constraint should always result in a constraint (exception: property 9).

**(8) Asymptotic bounds** This property states that the addition of two soft constraints with strengths less than 1 must not result in a hard constraint. Therefore, the state of an undoubtedly correct constraint can not be reached through the addition of soft constraints.

**(9) Conflicting constraints** An exception to closeness are hard constraints. The addition of a hard must-link constraint and a hard cannot-link constraint contradicts the definition of the constraint strength, as a strength of 1 signals that there is no doubt about the correctness of the constraint. Thus the addition should be undefined for this case. Instead, it has to be assured that such a situation does not occur.

Although some of these are actual the properties of an Abelian group,[3] the resulting methods are only of this kind when hard constraints are not considered. The reason for this is the exception introduced with the property of conflicting constraints. It should also be noted that this list of properties is not minimal. For example, the property of asymptotic bounds can be deduced from the properties of type equality, commutativity and conflicting constraints.

[3]These are the properties of inverse constraints, neutral constraint, associative property, commutative law and closeness.

Nevertheless, this list can be seen as a framework for the addition of soft constraints. If, however, one of the rules which are specified in the particular methods contradicts one of the general rules stated in the properties, then the rule of the particular method is applied.

**Maximum method** The maximum method considers only the constraint with the highest strength. Therefore, if two constraints are summed up, the result equals the addend with the higher strength. The maximum method serves as a baseline method for soft constraint addition, as it only considers one part of the present person attribute information. We denote the use of the maximum method by $+_{\max}$.

Although the idea behind this method is rather simple, it leads to a direct contradiction within the desired properties; at least one of the properties neutral constraint, inverse constraint and associative property and commutative law must be violated. We can prove this claim by an example: Consider three constraints, $c_=(d,d',s)$, $c_{\neq}(d,d',s)$ and $c_=(d,d',s')$ with $s > s' > 0$. Because of the properties of the neutral and the inverse constraint,

$$\begin{aligned}&(c_=(d,d',s) +_{\max} c_{\neq}(d,d',s)) +_{\max} c_=(d,d',s')\\ &= c_{\sim}(d,d',0) +_{\max} c_=(d,d',s')\\ &= c_=(d,d',s')\end{aligned}$$

holds. However, since the result of the maximum method is the addend with the higher strength, also

$$\begin{aligned}&c_=(d,d',s) +_{\max} (c_{\neq}(d,d',s) +_{\max} c_=(d,d',s'))\\ &= c_=(d,d',s) +_{\max} c_{\neq}(d,d',s)\\ &= c_{\sim}(d,d',0)\end{aligned}$$

holds. This leads to a direct contradiction since the associative property demands that

$$\begin{aligned}&(c_=(d,d',s) +_{\max} c_{\neq}(d,d',s)) +_{\max} c_=(d,d',s')\\ &= c_=(d,d',s) +_{\max} (c_{\neq}(d,d',s) +_{\max} c_=(d,d',s'))\end{aligned}$$

The proposed maximum method violates the property of the inverse constraint. More specifically, we introduce a threshold $\theta$ assigned to each "neutral" constraint and denoted by $c_{\sim}(d,d',0)_\theta$. If a must-link constraint and cannot-link constraint with the same strength $s$ are summed up, the result will be a neutral constraint with a threshold of $\theta = s$. Then, if a must-link or cannot-link with a strength $s'$ with $s' \leqslant s$ is added, the result will be again the neutral constraint. If, however, $s' > s$, then the result will be the must-link/cannot-link constraint. More formally,

$$\begin{aligned} c_{\sim}(d,d',0) &\equiv c_{\sim}(d,d',0)_0 \\ c_{=}(d,d',s) +_{\max} c_{\sim}(d,d',0)_\theta &= \begin{cases} c_{=}(d,d',s) & \text{if } s > \theta \\ c_{\sim}(d,d',0)_\theta & \text{if } s \leqslant \theta \end{cases} \\ c_{\sim}(d,d',0)_\theta +_{\max} c_{\sim}(d,d',0)_{\theta'} &= \quad c_{\sim}(d,d',0)_{\max(\theta,\theta')} \end{aligned}$$

Therefore, if the maximum method is applied in the previously stated prove, the result will be $c_{\sim}(d,d',0)_s$. The threshold $\theta$ is, however, only considered when the maximum method is applied. In all other cases, the neutral constraints with thresholds are used as normal neutral constraints.

Finally, constraint addition by employing the maximum method is defined for the remaining cases by

$$\begin{aligned} c_{=}(d,d',s) +_{\max} c_{=}(d,d',s') &= \quad c_{=}(d,d',\max(s,s')) \\ c_{=}(d,d',s) +_{\max} c_{\neq}(d,d',s') &= \begin{cases} c_{=}(d,d',s) & \text{if } s > s' \\ c_{\sim}(d,d',0)_s & \text{if } s = s' \\ c_{\neq}(d,d',s') & \text{if } s < s' \end{cases} \end{aligned}$$

**Multiplication** Unlike the maximum method, the *multiplication* method satisfies all desired properties. Furthermore, it tries to incorporate all information, not only the constraints with the highest strength. Instead, the constraint strength converges to 1 by adding constraints of the same type.[4] Let $+_{\text{mul}}$ denote the application of the multiplication method. Then, the addition of constraints via the multiplication method is defined by the following rules:

$$\begin{aligned} c_{=}(d,d',s) +_{\text{mul}} c_{=}(d,d',s') &= \quad c_{=}(d,d',1-(1-s)\cdot(1-s')) \\ c_{=}(d,d',s) +_{\text{mul}} c_{\neq}(d,d',s') &= \begin{cases} c_{=}(d,d',1-\frac{1-s}{1-s'}) & \text{if } s > s' \\ c_{\sim}(d,d',0) & \text{if } s = s' \wedge s \neq 1 \\ c_{\neq}(d,d',1-\frac{1-s'}{1-s}) & \text{if } s < s' \end{cases} \end{aligned}$$

As can be seen from the equations, the addition of a hard must-link and a hard cannot-link would results in a division by zero. Therefore, such as stated by the property of conflicting constraints, the addition remains undefined for this case.

An example for the calculation of the maximum and the multiplication method is given in Figure 3.3.

[4]If only constraints of one type are considered, then the sum of single strengths is calculated exactly as the joint probability of independent and not mutually exclusive events. There is, however, no similarity of the multiplication method to probability calculation if must-link constraints are added to cannot-link constraints.

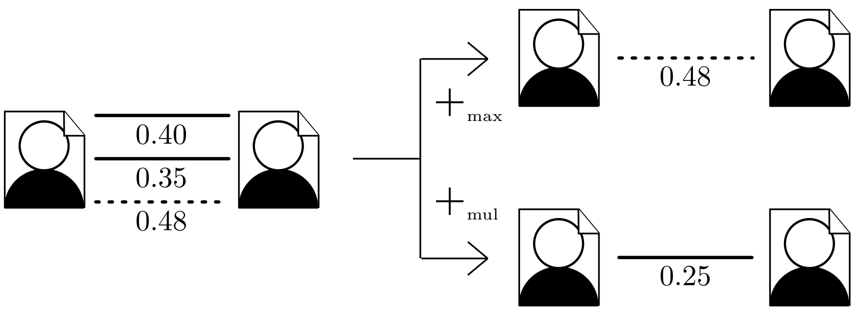


**Figure 3.3:** An example of the addition of two must-link constraints (solid line) and one cannot-link constraint (dotted line). The constraint strengths are shown below the lines. In the case of the multiplication method, the strength of the resulting must-link constraint is calculated by

$$1 - \frac{(1-0.4)\cdot(1-0.35)}{(1-0.48)} = 1 - \frac{0.39}{0.52} = 1 - 0.75 = 0.25$$

### Constraint Addition Within Attributes

The previously discussed methods for constraint strength addition are now applied to add up all soft constraints for each pair of web pages. To this end, the constraints $Con_a(d, d')$ as defined by Equation 3.1 have to be added up. Two constraints result from each person attribute and each pair of web pages: one must-link and one cannot-link constraint. The must-link constraint is created by summing up all constraints in $Con_a(d, d')$ by applying either the maximum or the multiplication method. This is also done for the cannot-link constraint. In this case, however, the strength is inverted by setting $s$ to $(1-s)$. Therefore, the strength of the cannot-link constraint is inversely related to the strength of the must-link constraint.

It should be noted, that it is possible that multiple pairs of person attribute values match with the highest strength for a single pair of web pages. Considering the motivation for the multiplication method, the more strong clues exist, the higher the strength should be. In order to still keep the strength within the bounds of the soft constraint strength of $[0, 1]$, the constrained strengths calculated in the matching-step are scaled by 0.5 as they are added up. Thus, 0.5 is the highest possible strength for one person attribute value and one person attribute. Moreover, this scaling prevents situations in which a hard must-link is added to a hard cannot-link constraint. Note that, since information which implies a grouping is handled differently from information which implies a separation of web pages, this scaling is performed at different steps. In order to allow a direct comparison of the multiplication and maximum method, the scores are also scaled for the maximum method by 0.5.

Since the addition of the constraints is independent from the selection of the method for addition, a general notation for addition (+) and for sums ($\sum$) is utilized. The within-attribute strengths for must-links ($\Sigma_=$) and cannot-links ($\Sigma_\neq$) are then defined by:

$$\Sigma_=(Con_a(d,d')) = \textit{strength-of}\left(\sum_{c_=(d,d',s)\in Con_a(d,d')} c_=(d,d',s/2)\right)$$

$$\Sigma_\neq(Con_a(d,d')) = \frac{1}{2}\cdot\left(1 - \textit{strength-of}\left(\sum_{c_=(d,d',s)\in Con_a(d,d')} c_=(d,d',s)\right)\right)$$

with *strength-of*$(c_=(d,d',s)) = s$.

Until this point, the clues from all person attributes are treated equally. But, as shown in Section 2.2.2 on page 16, the probability that a constraint is correct depends on the person attribute the constraint originated from. Furthermore, the probabilities differ for must-link and cannot-link constraints. Therefore, two *confidence factors* $\gamma \in [0,1]$ are applied to the within-attribute strengths of each attribute $a$: $\gamma_{a,=}$ to $\Sigma_=$ and $\gamma_{a,\neq}$ to $\Sigma_\neq$. When a confidence factor is close to 0 the effect of the within-attribute strength on the between-attribute strength is low. If $\gamma$ is 0, no constraints will be generated. A value between 0 and 1 is used if the person attribute information can be a supporting clue, but the error probability is too high to imply clustering decisions from it. Moreover, the confidence factors are a method of weighing out the information from different attributes. Both constraints are then again combined for the within-attribute constraint $c_a(d,d')$ of the pair of web pages $d$ and $d'$ for the person attribute $a$:

$$\begin{aligned} c_a(d,d') &= c_=(d,d',\gamma_{a,=}\cdot\Sigma_=(Con_a(d,d'))) \\ &\quad + c_\neq(d,d',\gamma_{a,\neq}\cdot\Sigma_\neq(Con_a(d,d'))) \end{aligned}$$

We assume that confidence factors are especially useful when the multiplication method is applied. In this case, the factors can be directly adjusted for multiple matches. For example, there might be several referents sharing an occupation. The number of referents sharing multiple occupations, however, is expected to be much lower. In the case of the email person attribute, however, no two referents should share an address. In order to account for this difference, a confidence factor can be applied to must-link constraints generated from the occupation person attribute. In detail, $n$ exactly matching person attribute values result in a constraint with strength $1-(1-0.5)^n = (2^n-1)/(2^n)$. Therefore, a confidence factor of $(2^{n-1})/(2^n-1)$ can be applied to adjust for $n$

matches. If, for example, $\gamma_{occupation,=}$ is set to 4/7 and $\gamma_{email,=}$ to 1, then 3 exactly matching occupations result in the same strength as one matching email address. It should be noted, however, that this calculation is only correct if exact matching is applied, but can still serve as a guideline in the case of soft matching.

#### Constraint Addition Between Attributes

After the must-link and cannot-link constraints for each person attribute and pair of web pages are generated, they are further added up to create a final soft constraint for each pair of web pages. Once more, either the maximum or the multiplication method is employed. The between-attribute constraint $c$ is calculated by summing up the within-attribute constraints for each person attribute $a$:

$$c(d, d') = \sum_a c_a(d, d')$$

The complete procedure for the addition of the similarity scores is shown in Figure 3.4. As depicted in the figure, no information is drawn from the absence of person attribute values. If, however, pairs of person attribute values exist for a person attribute, a must-link as well as a cannot-link constraint are created and then added up. No constraint will be generated if there does not exist a pair of values for any person attribute.

### 3.2.3 Transitive Closure, Entailment and Conflicts

Finally, a constraint set is created from the soft constraints. First, all soft constraints with a strength below a threshold $\theta_c$ are removed while conflicts are resolved. Then, the remaining constraints are converted to hard constraints under consideration of the transitivity and entailment. By applying this procedure, the constraints generated in the prior steps are either kept or sorted out. It should, however, be noted that it would also be possible to employ a method for the incorporation of the soft constraints into the employed similarity metric. This would leave the final decision of grouping or separating each pair of web pages to the clustering algorithm, which might or might not be advantageous for the referent disambiguation quality. Nevertheless, it would certainly make it more difficult to evaluate the generated constraints independent of a clustering algorithm (cf. Section 3.3).

The transitive closure as well as the entailment property of hard constraints are now further detailed. The closure leads to the notion of the *feasibility* of a constraint set. In order to make a constraint set feasible, conflicting constraints have to be identified and resolved. Two methods for conflict resolution are

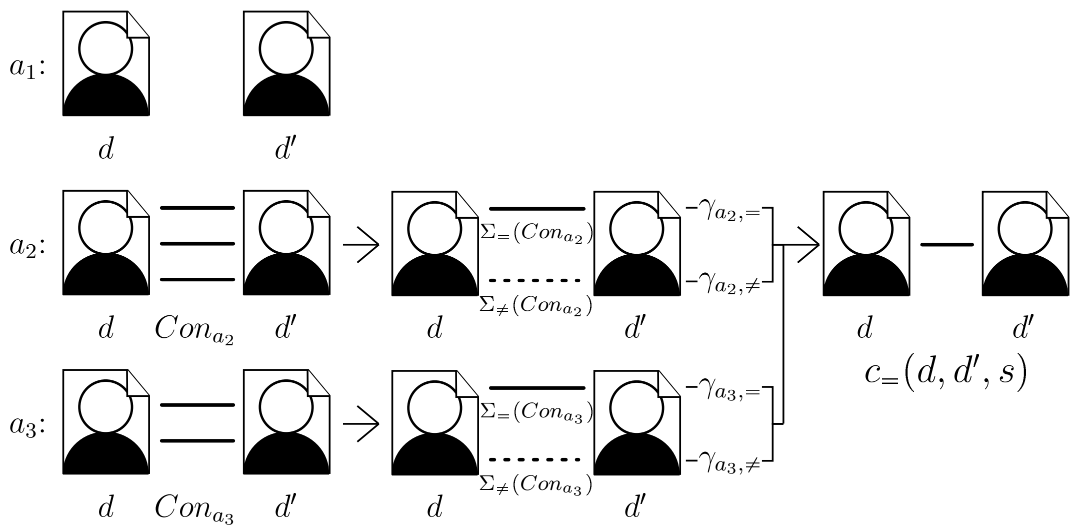


**Figure 3.4:** Example for an addition of similarity scores for one pair of web pages and three person attributes with zero, three and two person attribute value pairs respectively. The result—here the must-link constraint $c_=(d, d', s)$—can either be a must-link, a cannot-link or a neutral constraint.

proposed: (1) removing the conflicting cannot-link constraints and (2) raising the soft constraint threshold.

## Transitivity and Entailment

In order to compute the transitive closure of constraints, the single constraints which were created in the last steps have to be collected into a constraint set $Con_{\text{soft}}$:

$$Con_{\text{soft}} = \{c(d, d') : d \in D \wedge d' \in D\backslash\{d\}\}$$

which can then be converted into the two sets of hard constraints required by the algorithms:

$$\begin{aligned} Con_= &= \{c_=(d, d') : (c_=(d, d', s) \in Con_{=,\text{soft}}) \wedge (s \geqslant \theta_c)\} \\ Con_{\neq} &= \{c_{\neq}(d, d') : (c_{\neq}(d, d', s) \in Con_{\neq,\text{soft}}) \wedge (s \geqslant \theta_c)\} \end{aligned}$$

with $\theta_c \in [0, 1]$ as a confidence threshold for the constraint strength and $Con_{=,\text{soft}}$ and $Con_{\neq,\text{soft}}$ being the must-link and cannot-link constraints of $Con_{\text{soft}}$ respectively.

Considering the constraints of multiple pairs of web pages, further constraints can be deduced from some combinations of constraints [Wagstaff, 2002]. The two constellations of constraints in which this is the case are shown

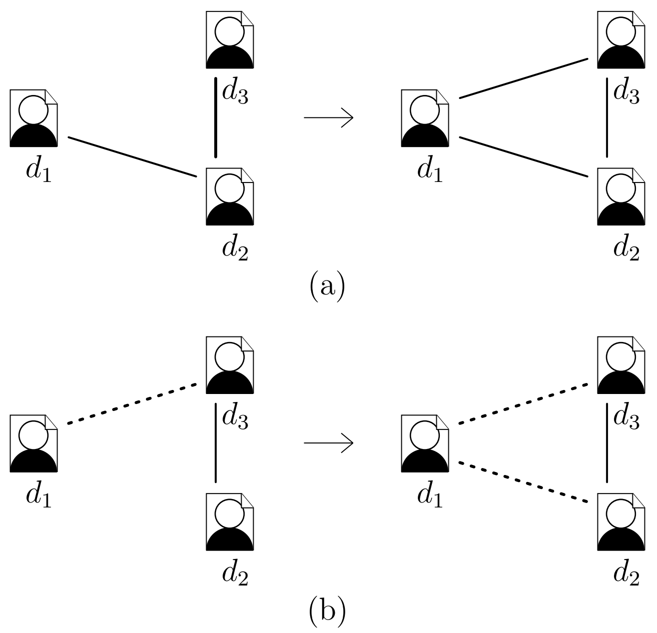


**Figure 3.5:** The two situations in which an additional constraint can be derived from other constraints: (a) transitivity of must-link constraints and (b) entailment property of must-link and cannot-link constraints.

in Figure 3.5 (a) and (b). In (a), $d_2$ has to be in the same cluster as $d_1$ and $d_3$. Hence, $d_1$ has also to be in the same cluster as $d_3$ (*transitivity*). Figure 3.5 (b) shows the situation in which a must-link and a cannot-link constraint are connected with the same web page: $d_2$ is connected by a must-link constraint with $d_3$, which must not be in the same cluster as $d_1$. Therefore, $d_1$ must not be in the same cluster as $d_2$ (*entailment*).

## Conflict Resolution

The transitive closure is, however, not considered during the creation of the constraints. Therefore, it is possible that conflicts arise. Consider again the situation shown in Figure 3.5, but this time with the constraints on the left part of (a) combined with those on the left part of (b):

$$Con_{=} = \{c_{=}(d_1, d_2), c_{=}(d_2, d_3)\} \qquad Con_{\neq} = \{c_{\neq}(d_1, d_3)\}$$

When the transitive closure is computed for this situation, $c_{=}(d_1, d_3)$, $c_{\neq}(d_1, d_2)$ and $c_{\neq}(d_2, d_3)$ are added to the constraint sets. As can be seen, this leads to a contradiction, as all pairs of web pages are required to be in the same as well as in different clusters.[5] Thus, these constraint sets are not *feasible* [Davidson

[5]Requiring a pair to be in same as well as in different clusters is not a contradiction if multiple-referent web pages are considered, but the disambiguation is left to the clustering

and Ravi, 2005].

In general, conflicts arise if the two web pages connected by a cannot-link constraint are also connected by a *path* of must-link constraints. A path of must-link constraints is a list in which each must-link constraint shares a connected web page with the previous, and the other web page with the following must-link constraint. To cause a conflict, a cannot-link constraint must exist which shares a web page with the first, and the other web page with the last must-link constraint of the path. Thus, the path of must-link constraints forms a *cycle* in conjunction with the cannot-link constraint. To resolve such a conflict, at least one of the constraints of the cycle has to be removed from the constraint sets. We continue now with the proposal of two strategies for conflict resolution: (1) taking must-link constraints and (2) raising threshold. After all conflicts are resolved, the two constraint sets can be passed to a constrained clustering algorithm (cf. Section 3.1.2).

**Taking Must-link Constraints** A simple solution is to always remove the cannot-link constraints of a cycle. As reasoned in Section 2.2.2 on page 16, extraction errors can more easily result in an incorrect cannot-link constraint than in an incorrect must-link constraint. Therefore, it is deemed more likely that the cannot-link constraint is wrong, and the path of must-link constraints is correct. An example application of this strategy is shown in the upper part of Figure 3.6. This strategy is considered a trivial solution and serves as a baseline for more sophisticated strategies.

**Raising Threshold** The second strategy utilizes the constraint strengths that are assigned to each constraint; if a conflict is detected, the constraint which has the lowest strength is removed. Unlike the first method, this strategy thus accounts for differences in constraint strength. An example of the application of this strategy is shown in the lower part of Figure 3.6.

## 3.3 Measures for Constraint Set Evaluation

In the last section, a framework for the generation of instance-based constraints from person attribute values was proposed. Furthermore, alternative methods were introduced for the single steps within the framework. Although some advantages and disadvantages of the different methods can be already derived

algorithm (cf. Section 2.1.3 on page 10). In this case, the illustrated situation suggests that the web pages concern common referents, as well as different ones. The transitive closure of the constraints, however, can not be calculated under this consideration.

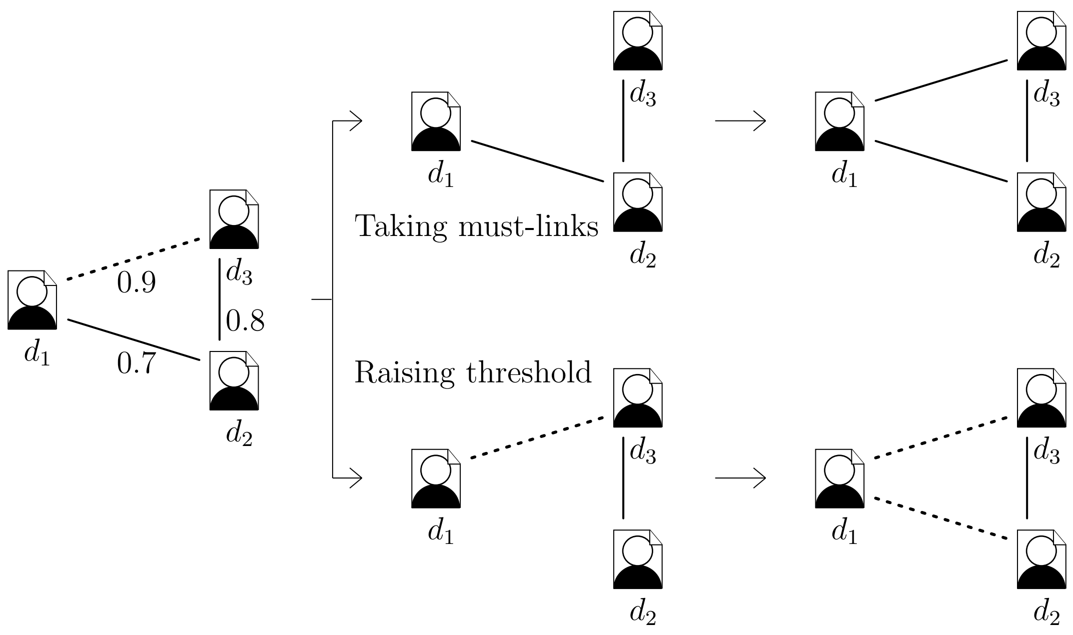


**Figure 3.6:** A situation with conflicting constraints is shown on the left. Furthermore, the two proposed strategies for resolving the conflict are illustrated: taking the must-link constraints (top) and raising the threshold (bottom).

from their definitions, it remains unclear if the advantages outweigh the disadvantages. To this end, an evaluation metric is required. For each possible constraint set, an evaluation metric defines a score which corresponds to the metric-specific quality of the set. Therefore, if an appropriate metric is chosen, it is possible to algorithmically compare different methods and different parameter settings.

Although various evaluation metrics exist for clustering tasks, to the best of our knowledge, none was proposed for the evaluation of the correctness and coverage of constraints. Most current and past research in constrained clustering concerns only constraints which are known to be correct. In these cases, either the constraints are directly provided by the user [Basu et al., 2004]; or they are generated by fail-safe rules [Wagstaff, 2002]. Nevertheless, two metrics were proposed by Davidson et al. [2006] for sets of correct constraints. These metrics were developed, because experiments showed that different constraint sets affect clustering quality differently. This was observed, even if the same number of constraints was applied and all constraints were undoubtedly correct. The proposed metrics are *informativeness* and *coherence*.

**Informativeness** The informativeness of a constraint set corresponds to the amount of constraints which are unintuitive with respect to the bias of the clustering algorithm.

**Coherence** The coherence considers the constraints from the point of view of the similarity metric. If must-link constraints are seen as space-contracting and cannot-link constraints as space-dilating, then incoherent constraints disagree on the dimensioning of the space in the view of the similarity metric.

As shown by Davidson and Ravi, constraint sets should have high coherence with the similarity metric while being informative for the clustering algorithm. Unfortunately, it is not trivial to define coherence for the cosine similarity, which we employ in our experiments. We assume, however, that incorrectness of the constraints has a greater effect on clustering quality than coherence.

In Section 3.3.1, constraint informativeness is considered in the context of constraints generated from person attribute values. After that, the well-known metrics precision and recall are adjusted for constraint sets in Section 3.3.2.

## 3.3.1 Constraint Informativeness

Constraints are employed to adjust the bias of the clustering algorithm to the current problem. An *informative* constraint will modify the bias, such that clustering decisions can be made which would contradict the unmodified bias. Other constraints, which also would have been satisfied if they were not enforced, affect the clustering much less, if at all. The *informativeness* I of a constraint set is then defined by Davidson et al. [2006] as the fraction of constraints which are informative (i.e., which are *not* satisfied by the clustering):

$$\mathrm{I}(Con, \mathcal{C}) = \frac{1}{|Con|} \cdot \left( \sum_{c_=(d,d') \in Con_=} (1 - \delta_{C(d),C(d')}) + \sum_{c_{\neq}(d,d') \in Con_{\neq}} \delta_{C(d),C(d')} \right)$$

with $\delta_{C(d),C(d')}$ being 1 if $d$ and $d'$ are in the same cluster, and 0 otherwise.

Davidson et al. employ informativeness to select one of multiple available constraint sets for clustering. Informativeness can, however, also be applied in the evaluation of constraint generation; in order to be of greatest use for the clustering algorithm, the generated constraints have to be informative with respect to the algorithm. Therefore, it is important for a thorough analysis of the generated constraints to consider also the informativeness of the constraints. Nevertheless, informativeness is insufficient as sole evaluation criterion since it does not consider the correctness of the constraints. Consequently, even totally incorrect and misleading constraints can achieve high informativeness. Hence, we do not employ informativeness as it is defined, but use it only in conjunction with precision and recall.

### 3.3.2 Precision and Recall of Constraint Sets

In the case of constraints, it is possible to impose them on the clustering process and then evaluate the resulting clustering. Since the reason for the creation of the constraints is to increase clustering quality, this provides a metric for measuring constraint set quality with respect to the web people search task. Nevertheless, an evaluation metric which is independent of the employed clustering algorithm is useful. In order to report reproducible results, the complete algorithm for web page clustering has to be made available, such that it can be used by others for constraint evaluation. Moreover, clustering algorithms might benefit differently from dissimilar constraint sets. Thus, the score of the evaluation metric looses validity if a different clustering algorithm is applied.

Precision and recall are the most basic evaluation metrics in information retrieval [Manning et al., 2008]. They are applied in tasks in which single *instances* have to be assigned to one of multiple pre-defined *classes*. Also, an algorithm for the task might choose not to classify an instance, because not enough evidence exists for neither class. In the case of constraint sets for web people search, an instance is a pair of web pages. The two available classes are must-link constraints and cannot-link constraints. If not enough evidence exists, the pair can be left unconstrained. Then, the definitions of precision and recall can be adapted from the general definitions in Manning et al. [2008]; *precision* P is the fraction of the constrained pairs of web pages which are correctly constrained; *recall* R is the fraction of possible correct constraints which were created by the algorithm. More formally,

$$\mathrm{P} = \frac{|\text{correctly created constraints}|}{|\text{correctly created constraints}| + |\text{incorrectly created constraints}|}$$

$$\mathrm{R} = \frac{|\text{correctly created constraints}|}{|\text{correctly created constraints}| + |\text{missing constraints}|}$$

Thus, an algorithm which creates many constraints, but many of them incorrectly, has usually a high recall, but a low precision. On the other hand, an algorithm might create constraints only if there is much evidence for the separation or grouping of the web pages. This results in a high precision, but in a low recall.

In order to account for different clustering algorithms, the informativeness of the constraints can be considered when precision and recall are calculated. If the clustering F-Measure is used as an evaluation metric for an applied constraint set, it is intuitive to analyze the constraint set not as it is, but as it is used by the clustering algorithm. To this end, the uninformative constraints can be removed from the three constraint sets (correctly

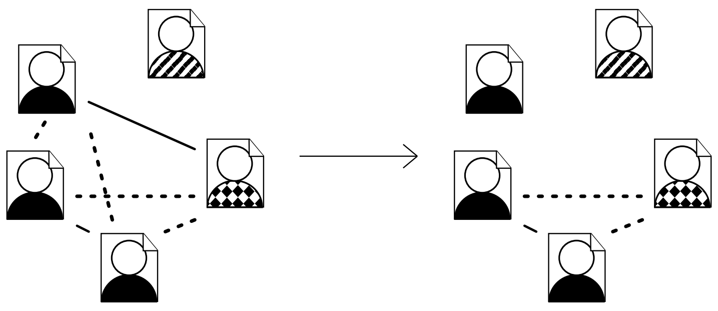


**Figure 3.7:** Creation of the set of correct constraints from the algorithm set: all must-link constraints between web pages which concern different referents and all cannot-link constraints between web pages which concern the same referent are removed.

created, incorrectly created and missing constraints) when precision and recall are calculated (*informative precision* IP and *informative recall* IR). Therefore, informative precision and recall consider only those correct, incorrect or missing constraints which are violated when the clustering algorithm is employed without constraints. Although these measures are useful for an analysis of how well constraints are incorporated into the clustering process, they are no longer independent of the clustering algorithm. Thus, they are not suited to evaluate the constraint set as such.

### Counting Constraints

In order to calculate precision and recall, the correct, incorrect and missing constraints have to be counted. To this end, three constraint sets are required:

1. The *reference* or *gold standard* set, which contains the correct constraint for each pair of web pages.
2. The *algorithm* set, which is the constraint set that is generated without knowledge of the correct solution.
3. The set of *correct* constraints, which is created by removing all incorrect constraints from the algorithm set (cf. Figure 3.7).

The set of correct constraints is thus the intersection of the reference and the algorithm set. As can be seen, the reference set requires the knowledge about the correct clustering of the web pages. Thus, precision and recall can only be calculated if this knowledge exists. While this is in general not true for user queries to a web people search system, special *test collections* or *corpora* of web pages are available which can be used instead (cf. Section 4.1.1 on page 50).

The different numbers of constraints can then be deduced from the three constraint sets in the following manner:

- The number of correct constraints is the number of constraints in the set of correct constraints.
- The number of wrong constraints is the number of constraints which have to be added to the set of correct constraints in order to create the algorithm set.
- Similarly, the number of missing constraints is the number of constraints which have to be added to the set of correct constraints in order to create the reference set.

It should be noted, however, that the constraints are not independent of each other. Some of the constraints can instead be derived from other constraints of the constraint set by the application of the rules of constraint transitivity and entailment (cf. Section 3.2.3 on page 38). Therefore, the constraints can be seen as single elements or as the representation of decisions to group or separate web pages. Both views are reasonable for measuring the quality of constraint sets.

The second view, which we refer to as *transitive count* (TC), does not reward the same decision multiple times. It favors constraint sets for which the constraints are scattered over the whole set. Scattered constraints will lead to several small groups during the application of must-link constraints in the clustering process, while concentrated constraints are likely do result in only a few, but large clusters. Having several small clusters at the start might be beneficial, as these clusters contain more information about the concerned referent as single web pages. Thus, an algorithm can make more founded decisions during the main part of the clustering process. On the other hand, decisions also have effects of different magnitudes on the clustering. The decision of merging or separating two big clusters has a greater impact on the final clustering than the same decision for small clusters. Therefore, it is plausible that such decisions should also have a greater effect on the evaluation score. This weighing of decisions is indirectly applied when the first view, which is called *absolute count* (AC), is considered.

An example of absolute and transitive count is given in Figure 3.8. Since must-link constraints can imply other constraints through constraint transitivity and entailment, these have to be applied first if calculating with transitive count. In this case, constraints also have to be applied one at a time, and are only counted if the constraint is not implied by the current constraint set. In the example, the recall in absolute count, $\mathrm{R}_{\mathrm{ac}}$, is $3/10 = 0.3$, while $\mathrm{R}_{\mathrm{tc}}$ is $2/5 = 0.4$. The precision can be calculated by employing the algorithm set.

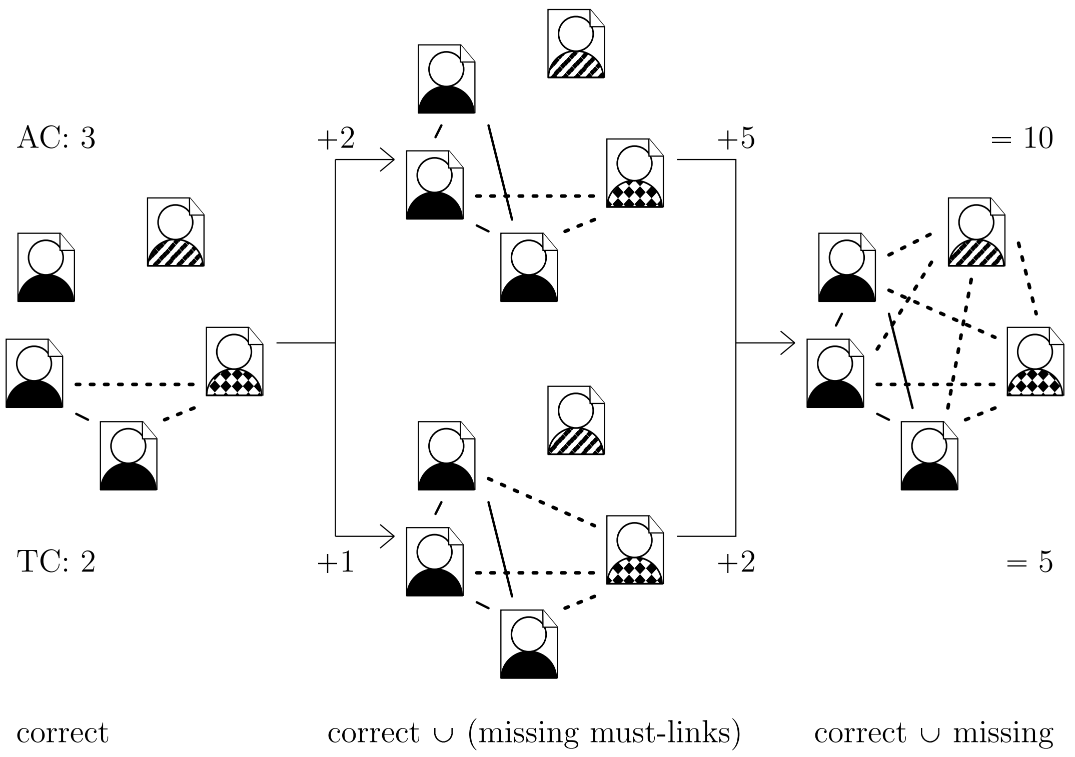


**Figure 3.8:** Calculation of recall by applying first the missing must-link and then the missing cannot-link constraints: (top) absolute count; (bottom) transitive count. Although it is not needed to apply the constraints in this order for absolute count, it is done this way in the example for comparison with transitive count.

For example in Figure 3.7, the absolute count precision $P_{ac}$ is $3/6 = 0.5$, while $P_{tc}$ is $2/3$.

## F-Measure

In order to combine precision and recall into a single metric, the F-Measure is employed again (cf. Equation 2.5 on page 12). For constraint sets, we argue that precision is of key interest, as incorrect constraints are even supposed to have a negative effect on the clustering quality. Thus, $\alpha$ should be chosen closer to 1 than to 0 in order to increase the significance of precision.

# Chapter 4

# Experiments

We address in this chapter the key questions which arise with the application of person attribute values as instance-based constraints in web people search. For this purpose, the proposed methods for the generation of constraints from person attribute values are empirically evaluated and analyzed. The WePS-2 corpus (cf. Section 4.1.1) is employed to perform evaluation of web page clustering. Furthermore, if not stated differently, the person attribute values of the gold standard of the WePS-2 corpus are utilized for the constraint generation in the experiments. In detail, we will give answers to the following research questions:

1. Which of the person attributes can be utilized to generate constraint sets which have both a high constraint set precision and a high constraint set recall? Can the differences in precision of cannot-link constraint sets be explained by the different values-per-referent ratios of the person attributes? Likewise, can the precision of must-link constraint sets be explained by the referents-per-value ratios of the person attributes?

2. Can constraint set precision and recall be weighed out by the application of soft matching of person attribute values, the utilization of the multiplication method and a change of the constraint strength threshold?

3. Can the performance of a clustering algorithm in the web people search task be improved by the enforcement of instance-based constraints? If so, can the magnitude of the gain in performance be explained by the precision, recall and informativeness of the employed constraint set?

4. Can a positive effect on web page clustering quality be achieved by the incorporation of constraints from different person attributes? Does the best constraint set with respect to clustering quality consist of must-link

constraints, of cannot-link constraints, or of both? In the latter case, how important is the confidence weighting of the constraints?

5. How is the quality of the generated constraints affected if algorithmically extracted person attribute values are employed instead of values extracted by human annotators?

This chapter is structured as follows. First, general remarks about the experiments are made in Section 4.1. Moreover, an overview of the employed parameters and evaluation methods is given. Then, the research questions are discussed in the context of the results of the experiments in Section 4.2.

## 4.1 Experimental Design

In web people search, evaluation is often performed by utilizing a web page collection and a gold standard [Artiles et al., 2010, Bagga and Baldwin, 1998]. Although the goal of web people search is to assist the user, a user study is rarely employed for evaluation. Compared to an algorithmic evaluation, a user study has the disadvantages of being (1) more costly in both time and money; (2) more difficult to reproduce; and (3) often restricted to few parameters of the web people search system. Therefore, an evaluation metric for measuring the quality of web people search systems is applied instead.

In order to objectively evaluate different methods of web people search, a publicly available web people search *corpus* is employed. Such a corpus contains several web people search *problems*, which are stated in the form of collections of web pages. Each collection contains web pages concerning different referents with the same target name. Additionally, the correct solution to each web people search problem (i.e., the correct grouping of the web pages by referents) is provided together with corpus. This solution is referred to as the *gold standard*. Furthermore, an evaluation metric is employed in order to compare solutions of web people search systems with the gold standard. For each possible solution, an evaluation metric defines a score which resembles the quality of the solution. The quality of the web people search system is then assumed as the average quality of the solutions for all problems. Similarly, the methods of constraint set generation we introduced in Section 3.2 (pp. 28 ff.) are evaluated by comparing the constraint sets which are generated for the single problems by the gold standard.

We use the WePS-2 corpus in our experiments (cf. Section 4.1.1), since it also features a list of human-extracted person attributes for each web page in the corpus, which is required for our experiments. In order to answer the previously stated research question, several experiments are performed. The

**Table 4.1:** The average ($\mu$) and the standard deviation ($\sigma$) of the number of referents ($|R|$) and web pages ($|D|$) for each target name in the WePS-2 corpus.

| Part of the corpus | $\mu_{\|R\|}$ | $\sigma_{\|R\|}$ | $\mu_{\|D\|}$ | $\sigma_{\|D\|}$ |
|---|---|---|---|---|
| English Wikipedia | 10.70 | 12.52 | 94.00 | 7.31 |
| ACL'08 | 14.20 | 9.84 | 81.60 | 12.99 |
| 1990 US Census | 30.30 | 14.71 | 80.20 | 12.02 |
| Complete corpus | 18.40 | 15.15 | 85.27 | 12.68 |

experiments differ in the employed parameters (cf. Section 4.1.2) as well as in the applied evaluation metrics (cf. Section 4.1.3).

It should be noted that if the reader has already a clear understanding of our proposed framework of constraint generation and is in general familiar with the field of information retrieval, these sections are not required in order to comprehend the results of the experiments. Thus, the reader might want to skip to Section 4.2 on page 56. However, the sections are helpful if the reader wants to get an overview of the methods we employ (with references to the detailed description) or wants to learn about the details of the conducted experiments. Furthermore, the utilized Pearson correlation coefficient and method of leave-one-out cross-validation, which the reader might be unfamiliar with, are explained in Section 4.1.3.

### 4.1.1 The WePS-2 Corpus

In our experiments, the test data of the second web people search workshop (WePS-2, [Artiles et al., 2009, Sekine and Artiles, 2009]) is used. The corpus is freely downloadable from the workshops website.[1] It consists of 3 444 web pages in English language which are the result of 30 personal name queries to *Yahoo!* web search. The pages are provided together with the URL and title, as well as with their snippet and rank in the search result list. The target names are taken from three sources (10 names each): (1) the English Wikipedia; (2) the names of members of the Programme Committee of the Association for Computational Linguistics' meeting in 2008 (ACL'08); (3) randomly composed first and last names by using the probabilities of the 1990 US Census data. The number of different referents for a target name varies between 1 and 56 referents [Artiles et al., 2009]. The average and the standard deviations of the number of referents and web pages for each of the three referent-groups are given in Table 4.1.

[1] `http://nlp.uned.es/weps/weps-2`

Manual annotation was used in order to create a gold standard for web page clustering and attribute extraction [Artiles et al., 2009, Sekine and Artiles, 2009]. A clustering or an attribute list identical to the gold standard is assigned the highest possible score. Some of the web pages, however, were discarded by the annotators and are therefore not used in the evaluation. A reason for discarding a web page was, for example, that it is even for the annotator unclear to whom the web page refers to. For the clustering task, 2 558 web pages (74.3% of all web pages) remained for the evaluation. From the 2 883 web pages which are used for the evaluation of attribute extraction systems, 2 421 (70.3%) have at least one attribute value. In total, the gold standard lists 11 253 attribute values [Sekine and Artiles, 2009]. Since all noise pages (cf. Section 2.1.3 on page 10) are discarded from the evaluation, we do not employ any method for the detection of noise pages in our experiments.

It should be noted that the annotators were allowed to assign a web page to multiple referents. This, however, was rarely used: only 12 of the 2 558 web pages for the evaluation (0.5%) concern multiple referents.[2] Therefore, although multiple-referent pages are considered by the employed evaluation metric (cf. Appendix A on page 83), we do not consider them during the grouping of web pages. Furthermore, we remove all person attribute values from multiple-referent pages since it is unclear to whom of the referents they belong.

### Web Page Pre-processing

Before clustering, the web pages of the WePS-2 corpus are transformed into feature vectors by (1) applying the Jericho HTML Parser;[3] (2) removing stopwords; (3) performing stemming; and (4) using the tf·idf weighting scheme. First, the textual content of the web pages of the WePS-2 corpus are extracted by the application of the Jericho HTML Parser library. Second, the textual content is processed as described in Section 2.1.1 on page 6. After the textual content is converted to lower case, we remove all words contained in the list of English stopwords of the AItools library[4] as well as all punctuation. Then, stemming is performed by applying the Snowball stemmer implementation[5]

[2]Nevertheless, web pages concerning multiple referents are likely to occur more frequently on the Internet than it is expressed in this numbers; in the corpus of the predecessor workshop, WePS-1 [Artiles et al., 2007], 7.1% of the web pages used in the evaluation are multiple-referent pages. This significant difference is likely due to the modified guidelines for discarding web pages in WePS-2, where genealogy pages are also removed from the evaluation. As a consequence, 25.7% of all web pages are discarded in the WePS-2 corpus, in contrast to only 15.1% in the WePS-1 corpus.

[3]`http://jericho.htmlparser.net`

[4]`http://aitools.de`

[5]`http://snowball.tartarus.org`

of Porter's stemming algorithm for the English language [Porter, 1997]. The resulting single words (*unigrams*) are used as the base unit for the vector space model. In order to improve the prediction of the document frequencies of the unigrams (cf. Section 2.1.1), we process the web pages of the similar WePS-1 corpus [Artiles et al., 2007] equivalently and compute the document frequencies based on all web pages in both corpora. Nevertheless, we employ only the unigrams that occur in the WePS-2 corpus for web page representation. Moreover, the unigrams are further reduced to only the 10 000 most frequent ones. This step was originally introduced to allow for metric learning algorithms, some of which require space quadratically growing in the number of terms ($\mathcal{O}(n^2)$). Within the last weeks of the work on this thesis we decided not to apply such algorithms and to concentrate on the methods we described in Section 3.1.2 on page 25. Unfortunately, not enough time remained to rerun our experiments with the full set of unigrams. However, only a very small negative impact on clustering quality was observed through this dimensionality reduction in the case of the unconstrained clustering algorithms. We therefore assume that also the impact on the clustering quality of the constrained algorithms is negligible.

#### Person Attribute Values

We employ the person attribute values of the WePS-2 attribute extraction gold standard [Sekine and Artiles, 2009] as well as those extracted by the PolyUHK attribute extraction system [Chen et al., 2009] (cf. Section 2.2.1 on page 14).[6] Both sets of person attribute values are extracted from the web pages of the WePS-2 corpus. Whereas, however, the gold standard person attribute values are extracted by human annotators, the second set of person attribute values are extracted by a computer algorithm.

The eight person attributes we describe in Table 2.1 on page 14 are employed in our experiments. Moreover, the person attribute values are preprocessed such as stated in Table 2.1.

### 4.1.2 Parameters

The employed parameters of the experiments are summarized and detailed in this section.

[6]We received the person attribute values, which Chen et al. [2009] submitted as their entry to the WePS-2 workshop, by sending a request to the authors. We would like to thank Chen et al. for this kind support of our work.

### Person Attributes and Constraint Types

We apply constraints of different types which resulted from the matching of the different person attributes. If not stated otherwise, we always employ the person attribute values of the WePS-2 corpus in our experiments. The person attribute values extracted by the PolyUHK system are only applied for a comparison with the WePS-2 *gold standard* attributes in Section 4.2.5. Furthermore, we compare the utility of the different person attributes for web people search. In this context, we apply constraint sets of constraints of one type (i.e., either *must-link* constraints or *cannot-link* constraints) and which were generated by considering only the values of one person attribute. We refer to such constraint sets as *single constraint sets*.

### Constraint Set Generation

The different methods for the generation of the constraints (cf. Section 3.2 on page 28) are compared in the experiments. The first of these methods is the matching of the person attribute values: either *exact* or *soft* matching is applied (cf. Section 3.2.1). The constraint strengths of the resulting soft constraints are then either added up by using the *maximum* or the *multiplication* method as they are defined in Section 3.2.2.

Moreover, the *constraint strength threshold* $\theta_c$ (cf. also Section 3.2.2) is applied to determine the minimum constraint strength of a soft-constraint such that it is employed as a hard constraint in web page clustering. If not stated otherwise, we use $\theta_c = 0.5$ which equals to one exact match—or mismatch in the case of cannot-link constraints—of person attribute values because of the scaling applied during constraint strength addition. If different constraint strength thresholds are used in an experiment, we refer to *all steps of the constraint strength threshold* as the series of thresholds from 0 to 1 in steps of 0.0625. This interval (1/16) is motivated by the constraint strength of multiple exact matches within one constraint type for the multiplication method; the constraint strengths $0.5000, 0.7500, 0.8750$ and $0.9375$ (or $1/2, 3/4, 7/8$ and $15/16$) are reached by $1, 2, 3$ and $4$ exact matches in person attribute values.

### Joint Application of the Constraint Types

Only if single constraint sets of different constraint types are combined, conflicts can occur. Thus, the different strategies for conflict resolution (cf. Section 3.2.3 on page 40) are first applied in the experiments for the joint application of the single constraint sets in Section 4.2.4. We distinguish the *taking-must-link-constraints* and the *raising-threshold* strategy. Moreover, person attribute weighting by confidence factors (cf. Section 3.2.2) is also only applied when different constraint types are employed. Therefore, we only use the confidence factors 0 (constraints are not applied) and 1 (constraints are applied) for the single constraint sets if not stated otherwise.

**Clustering**

Finally, the two constrained clustering algorithms single pass clusterer (*SPC*, cf. Algorithm 3.1 on page 27) and constrained single link hierarchical agglomerative clusterer (*HAC*, cf. Algorithm 3.2 on page 28) are employed in order to quantify the effect of the constraint sets on web page clustering. We always apply the WePS-2 corpus in web page clustering.

Both algorithms are parametrized with a similarity threshold $\theta_{\text{SPC}}$ or $\theta_{\text{HAC}}$ respectively. Through this threshold, the algorithms can be adjusted between creating a singleton cluster for each web page ($\theta > 1$, *one-in-one*) and clustering all web pages into a single cluster ($\theta = 0$, *all-in-one*). Similarity thresholds are applied in the range from 0 to 1 in steps of 0.025. For the sake of brevity we refer to these steps as *all steps of the similarity threshold*.

These algorithms are also applied in order to calculate the *informativeness* of constraint sets. In this case, the informativeness is calculated with respect to the clustering which is generated when the best global similarity threshold is applied. The best similarity threshold is determined by calculating the BCubed F-Measure (see Section 2.1.4 on page 11) for the clusterings which result from the application of all steps of the similarity threshold and selecting the threshold for which the highest F-Measure is achieved. The best similarity thresholds are 0.175 for the single pass clusterer and 0.300 for the hierarchical clusterer.

## 4.1.3 Evaluation

Our experiments are evaluated on three different levels: person attributes, constraint sets and web page clusterings. In each of the three cases, the evaluation scores are calculated by computing the evaluation score for each problem and then averaging these scores (*macro averaging*).

First, we evaluate the different person attributes by the employment of the referents-per-value and values-per-referent ratios we defined in Equation 2.6 on page 19. It should be noted that we defined the two ratios only for exact matching of person attribute values and only consider this case in our experiments.

Second, the constraint sets generated from the person attribute values are directly evaluated in terms of precision ($\text{P}_{\text{ac}}/\text{P}_{\text{tc}}$) and recall ($\text{R}_{\text{ac}}/\text{R}_{\text{tc}}$) by employing absolute (ac) or transitive count (tc) as defined in Section 3.3.2 on page 44. Moreover, informative precision ($\mathrm{IP}_{\mathrm{ac}}/\mathrm{IP}_{\mathrm{tc}}$) and recall ($\mathrm{IR}_{\mathrm{ac}}/\mathrm{IR}_{\mathrm{tc}}$, cf. also Section 3.3.2) are employed to account for the informativeness of the constraint sets with respect to an clustering algorithm. It should be noted that the informativeness of a constraint set is calculated by employing the best average clustering as it is detailed in Section 4.1.2. Additionally, the F-Measure (cf. Equation 2.5 on page 12) is utilized to combine precision and recall into a single evaluation metric.

For some experiments, the numbers of correct (TP, *true positives*) and incorrect constraints (FP, *false positives*) are shown instead of precision and recall in order to visualize the correctness and amount of the constraints in one chart. In this case, the number of informative correct (ITP) and informative incorrect constraints (IFP) are also shown as a part of the correct/incorrect constraints. If the number of correct and incorrect constraints are shown, always absolute counting is employed. It should be noted that in the case of correct and incorrect constraints the sum of the constraints for all 30 problems is shown and no averaging is applied.

Third, the clusterings which result from the enforcement of constraint sets on a clustering algorithm is evaluated by using the F-Measure of extended BCubed precision and recall (cf. Appendix A on page 83). We employ $F_{\alpha=0.5}$ for web page clustering such as in the WePS-2 workshop [Artiles et al., 2009]. Therefore, it is always referred to $F_{\alpha=0.5}$ of BCubed precision and recall if F-Measure is used in the context of web page clustering.

### Correlation

The correlation of two evaluation metrics is calculated by using Pearson's correlation coefficient. We show in the experiments that the first level of evaluation (person attribute values) is correlated with the evaluation of the second level (constraint sets). Furthermore, the second level is correlated with the third level (web page clustering).

The Pearson correlation coefficient $r$ is calculated for two variables. It is computed by dividing the covariance of the two variables by the product of the standard deviations of the variables. The result lies in the range from $-1$ to 1, with $r=1$ in the case of positive linear correlation, $r=-1$ in the case of negative linear correlation, and $r=0$ if the two variables are uncorrelated.

### Leave-one-out Cross-validation

In order to reduce the number of parameters of the experiments, leave-one-out cross-validation (*loo-cv*) is applied for the similarity threshold $\theta_{\mathrm{SPC}}/\theta_{\mathrm{HAC}}$ of the clustering algorithm and for the constraint strength threshold $\theta_c$. In leave-one-out cross-validation, the parameter setting for a problem is determined as the best parameter setting for all other problems.

More specifically, if leave-one-out cross-validation *is applied for one of the thresholds*, the threshold for each of the 30 web people search problems of the corpus is selected by considering only the 29 other web people search problems. First, the best threshold for each possible set of 29 of the 30 problems is computed as the threshold for which the highest average F-Measure for the 29 problems is achieved. Second, the 30 computed thresholds are each applied to the problem which was not considered in the calculation of the threshold. Third, the 30 F-Measure scores which result from the application of the thresholds are averaged. This average F-Measure score is then used as the result of the cross-validation.

If leave-one-out cross-validation is performed for both the similarity threshold and the constraint strength threshold, the best combination of both thresholds is computed for each possible set which contains 29 of the 30 problems. Therefore, the best pair of thresholds is chosen from all combinations of all steps of the constraint strength threshold (17 steps) and all steps of the similarity threshold (41 steps).

Leave-one-out cross-validation therefore resembles a realistic situation in which the best parameter for a particular problem is unknown, but an annotated corpus of *similar* problems exists.

## 4.2 Results and Discussion

The research questions stated at the beginning of Chapter 4 are now addressed. An answer to the questions on the suitability of the different person attributes for constraint generation is given in Section 4.2.1. If it is possible to weight out constraint set precision and recall by the utilization of soft matching and the multiplication method is discussed in Section 4.2.2. The application of single constraint sets in web page clustering is evaluated in Section 4.2.3. Constraint sets which are generated by employing different person attributes and constraint types are then applied in Section 4.2.4. Finally, Section 4.2.5 addresses the question of how clustering quality is affected if algorithmically extracted person attribute values are employed instead of manually extracted ones.

### 4.2.1 Precision and Recall of Single Constraint Sets

In order to answer the research questions under item 1, we generate constraint sets containing only constraints of one person attribute and one constraint type, and compute the precision and recall of these. Furthermore, the Pearson correlation coefficient is calculated for the values-per-referent ratios and

**Table 4.2:** Precision (P) and recall (R) of constraint sets of one type and generated from the values of one person attribute. Precision and recall are given in absolute (ac) and transitive count (tc).

| | Cannot-link constraints | | | | Must-link constraints | | | |
|---|---|---|---|---|---|---|---|---|
| Attribute | $\mathrm{P}_{\mathrm{ac}}$ | $\mathrm{P}_{\mathrm{tc}}$ | $\mathrm{R}_{\mathrm{ac}}$ | $\mathrm{R}_{\mathrm{tc}}$ | $\mathrm{P}_{\mathrm{ac}}$ | $\mathrm{P}_{\mathrm{tc}}$ | $\mathrm{R}_{\mathrm{ac}}$ | $\mathrm{R}_{\mathrm{tc}}$ |
| Affiliation | 0.747 | 0.747 | 0.092 | 0.394 | 0.956 | 0.963 | 0.060 | 0.119 |
| Date of birth | 0.989 | 0.989 | 0.002 | 0.021 | 0.933 | 0.937 | 0.015 | 0.053 |
| Birthplace | 0.773 | 0.773 | 0.003 | 0.024 | **1.000** | **1.000** | 0.005 | 0.031 |
| Email | 0.945 | 0.945 | 0.004 | 0.035 | 0.929 | 0.929 | 0.001 | 0.010 |
| Nationality | 0.750 | 0.750 | $\sim 0.0$ | 0.003 | **1.000** | **1.000** | 0.014 | 0.048 |
| Occupation | 0.658 | 0.658 | **0.115** | **0.425** | 0.790 | 0.868 | **0.097** | **0.177** |
| School | 0.870 | 0.870 | 0.005 | 0.034 | 0.903 | 0.929 | 0.008 | 0.031 |
| Third name | **0.996** | **0.996** | 0.016 | 0.067 | 0.743 | 0.812 | 0.007 | 0.016 |

cannot-link constraint set precision, as well as for the referents-per-value ratios and must-link constraint set precision. We expect that constraint sets with a high score in precision can be generated from the person attribute values of person attributes with a value close to 1 in the corresponding ratio.

In the experiment, constraints are generated by performing exact matching of the attribute values and applying the maximum method for constraint strength addition. Thus, a must-link constraint is created between two web pages if two values of the same person attribute match exactly, and a cannot-link constraint if none of the person attribute values match. Precision and recall, which are shown in Table 4.2, are then calculated from the resulting constraint sets by absolute or transitive count. It should be noted that no further constraints are created through constraint transitivity or entailment if only cannot-links are applied; therefore, the precision is the same for absolute and transitive count in this case.

The values in Table 4.2 are closely related to the values-per-referent and referents-per-value ratios given in Table 2.2 on page 20. Birthplace and nationality, the two attributes with the maximum referents-per-value of 1 are also the only attributes for which a must-link constraint set precision of 1 could be achieved. Occupation, on the other hand, is the attribute with the highest values-per-referents ratio (of 9.60) and also the lowest cannot-link constraint precision. Moreover, although the values calculated by transitive count are generally higher than those for absolute count, they are also closely related; absolute and transitive count precision of the must-link constraint sets are almost linearly correlated with $r = 0.986$.

The linear relationship between constraint precision and values-per-referent

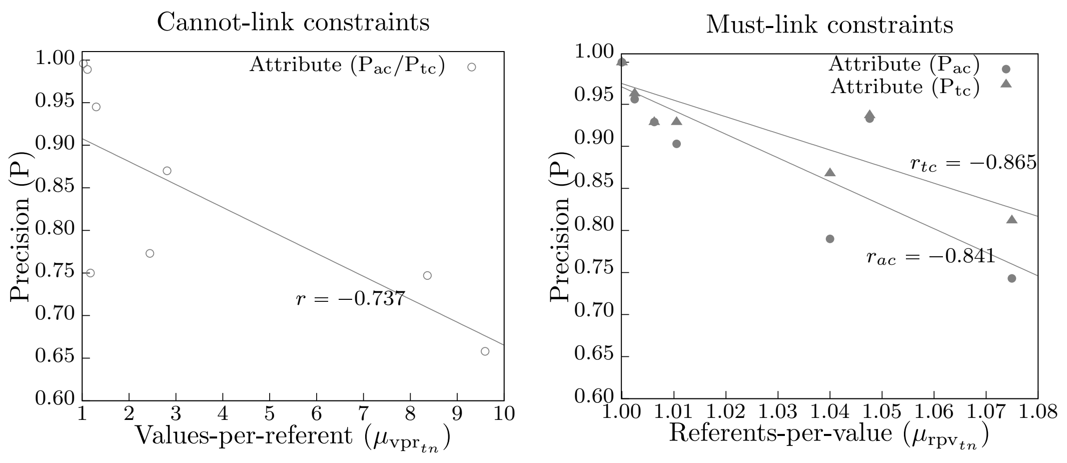


**Figure 4.1:** Correlation between the referents-per-value (values-per-referent) ratio and the precision of the cannot-link (must-link) constraint sets. The eight employed person attributes are represented as symbols. Precision is calculated by either applying absolute (ac) or transitive count (tc).

or referents-per-value ratio is shown in Figure 4.1. The elements of the scatterplots correspond to the single person attributes. In the case of must-link constraints (right), one element for absolute and one for transitive count precision is shown for each attribute. As implied by the values of the correlation coefficient $r$, which are shown in the figure, the values-per-referent/referents-per-value ratios of the person attributes are correlated with the precision of the constraint sets generated from the person attribute values. Hence, the two ratios can be accepted as measures of the suitability of an attribute for the generation of reliable cannot-link/must-link constraints.

### 4.2.2 Soft Matching of Person Attribute Values

In order to address the question under item 2, we repeat the experiments which are conducted for the first questions with different values for the constraint strength threshold $\theta_c$. We apply either soft matching instead of exact matching, the multiplication method instead of the maximum method, or both. We assume that the application of a lower constraint strength threshold in combination with soft matching will reduce the precision, but increase the recall of the constraint sets. On the other hand, we expect that a higher precision can be achieved at the cost of a lower recall if the multiplication method is employed in conjunction with a higher value for the constraint strength threshold.

### Decreased Constraint Strength Threshold

First, we will discuss if the recall of a constraint set can be increased by the application of soft matching and a decreased constraint strength threshold. Representative and exceptional results of the experiment are shown in Figure 4.2. It should be noted that the precision and recall for exact matching are the same as for soft matching with a threshold of 0.5. For cannot-link constraints, the date of birth person attribute is shown as a representative example. By reducing the threshold (i.e., reading from right to left in the graph), recall increases slowly, but precision decreases. In the case of date of birth, the first increase of recall at 0.375 shows that the corpus contains referents for which different days are given for date of birth. However, more incorrect decisions are made since the precision drops. Similar relations of precision, recall and constraint strength threshold are also observed for the person attributes affiliation, birthplace, email and occupation. In the case of nationality or third name, precision and recall are unaffected by a change of the threshold. This is because person attribute values exist for only few referents (nationality) or because the person attribute values of the referents are very different (third name).

An exceptional behaviour for cannot-link constraints is observed for the school attribute. In this case, both precision and recall can be increased by a small decrease in the constraint strength threshold. Therefore, of the additional constraints which result from a smaller threshold, even a higher percentage than in the constraint set which results from exactly matching attributes are correct. We assume that the relative high increase in recall (from 0.005 to 0.010 in absolute count) can be explained by the fact that some words occur very frequently in different attribute values of this person attribute (e.g., “School” or “University”). Therefore, the names of different educational institutions often have a small similarity when compared with the soft-tf·idf metric.[7] Hence, if exact matching is applied, no cannot-link constrained is created. We have, however, not yet found an answer for explaining the increase in precision.

In the case of must-link constraints, occupation is employed as a representative person attribute. Similar to the situation for cannot-link constraints, recall increases slowly while precision decreases. For precision, saturation curves which are similar to the one of occupation can be observed for the person attributes affiliation, birthplace, date of birth, email, occupation and (partly) school. Analogous to the situation for cannot-link constraints, precision and recall of the person attributes nationality and third name are independent of

[7]The similarity is small since the effect of the frequent words is reduced by the inverse document frequency weighting of the metric.

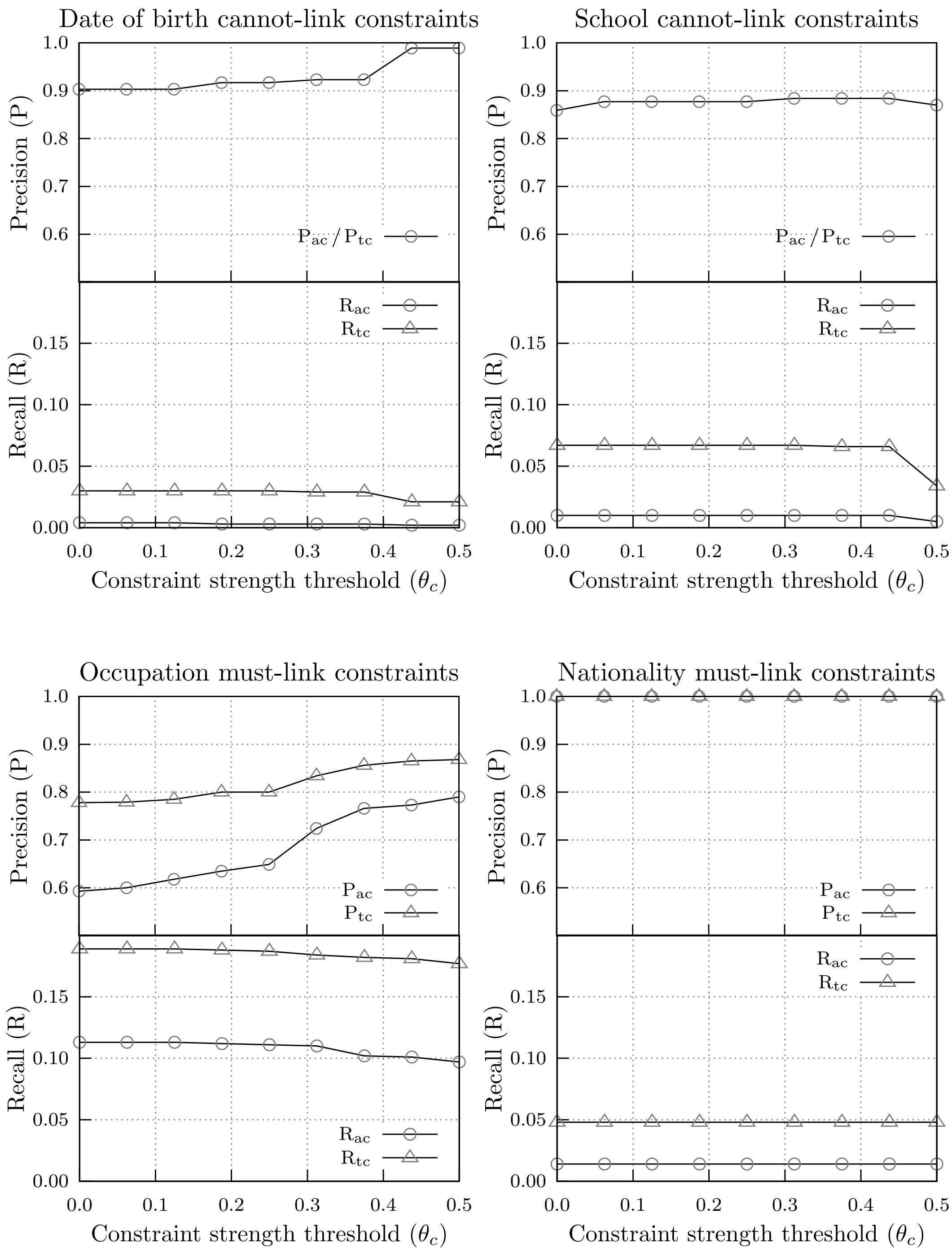


**Figure 4.2:** Precision (P) and recall (R) of constraint sets generated from four different person attributes in dependence on the constraint strength threshold. The constraint sets are generated by employing the maximum method and soft matching. Precision and recall are calculated by applying either absolute (ac) or transitive count (tc).

the employed threshold.

Thus, precision and recall can be outweighed by the application of soft matching and a lower constraint strength threshold *for some person attributes*. However, since the effect on recall is rather small in most cases, only small adjustments can be made. Quite counterintuitively, a pure positive is observed in the case of the school attribute.

#### Increased Constraint Strength Threshold

An increased constraint strength threshold, which requires multiple matches for one person attribute, is applicable to only few of the considered person attributes: affiliation, occupation and school. Web pages contain only in rare cases multiple values for one of the other person attributes. Moreover, within our framework it is not possible that a cannot-link constraint generated from a single person attribute has a strength above 0.5. Therefore, we only consider an increased threshold for the three mentioned person attributes.

Precision and recall for the application of the maximum or the multiplication method and soft or exact matching are shown for the person attribute occupation in Figure 4.3. Similar results are, however, observed for all of the three considered person attributes, although it should be noted that recall declines faster in the case of the school attribute. Again, the combination of the methods exact matching and maximum constraint strength addition is represented by the value of maximum method/soft matching for a threshold of 0.5. Since a constraint strength above 0.5 can not be achieved in terms of the maximum method, the recall of the maximum method is zero for higher thresholds. We want to note that generally a higher recall is achieved when soft matching is employed, but also a lower precision. The application of soft matching, however, allows for a more continuous finetuning of precision and recall. In the case of the occupation method, a good value for the constraint strength threshold is 0.8125, with a precision of 0.968 and a recall of 0.043. In comparison with the email attribute ($P_{ac} = 0.945$, $R_{ac} = 0.004$), a higher precision and a recall ten times as high is achieved.

### 4.2.3 Constrained Web Page Clustering

We address the questions under item 3 by enforcing the constraint sets which are generated for the experiment of the first questions on the two constrained clustering algorithms. The extended BCubed F-Measure is employed as a measure of the clustering performance. The F-Measure score which is achieved by the constrained clustering algorithms is then compared to the score which is achieved without the application of constraints. We expect that a high

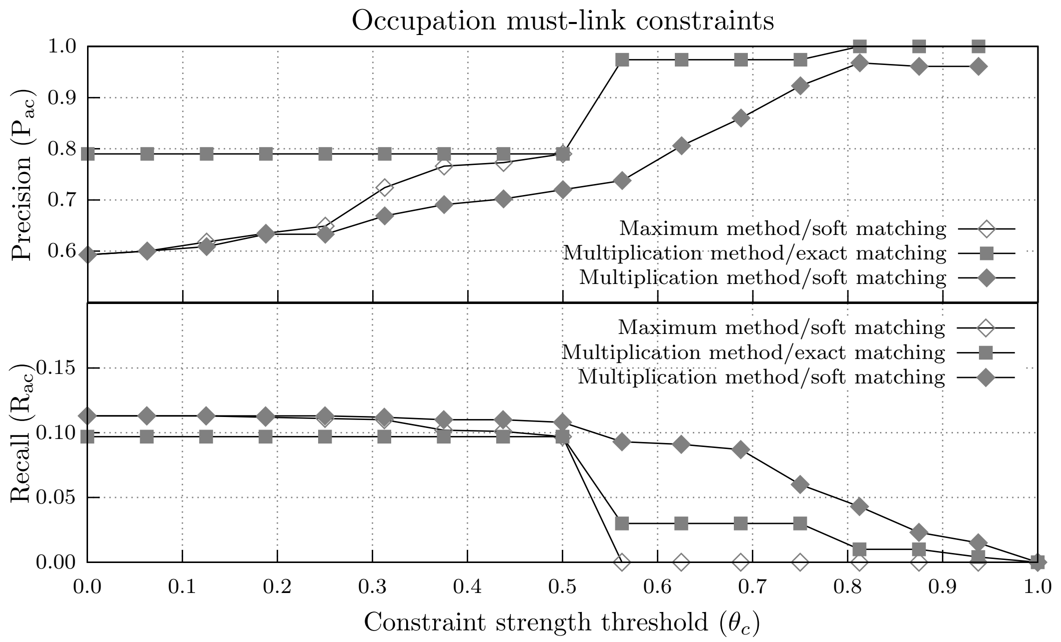


**Figure 4.3:** Precision (P) and recall (R) of the must-link constraint set in dependence on the constraint strength threshold. The constraint sets are generated by employing either the maximum or the multiplication method and either exact or soft matching. Precision and recall are calculated in absolute count. It should be noted that precision is undefined if no constraints are generated (and recall is 0).

constraint set precision is required for an increase in the clustering F-Measure. Furthermore, we assume that the informativeness of the constraint sets is also of key importance in this regard. Finally, it is expected that no significant difference exists between the change of the performance of the single pass clusterer and the hierarchical clusterer, since the two algorithms enforce the constraints in a similar way.

## Unconstrained Clustering Performance

Figure 4.4 shows the BCubed F-Measure for the two unconstrained clustering algorithms introduced in Section 2.1.2: single pass clustering and hierarchical agglomerative clustering. If the similarity threshold is 0, then both clustering algorithms form one cluster which contains all web pages (*all-in-one*). If the threshold is greater than 1,[8] then both algorithms cluster each web page into a singleton cluster (*one-in-one*). As it is shown in the graph, both algorithms are able to improve on these two *baselines* when an appropriate threshold is

[8]Only web pages which are identical with respect to the retrieval model are clustered together if the threshold is equal to one.

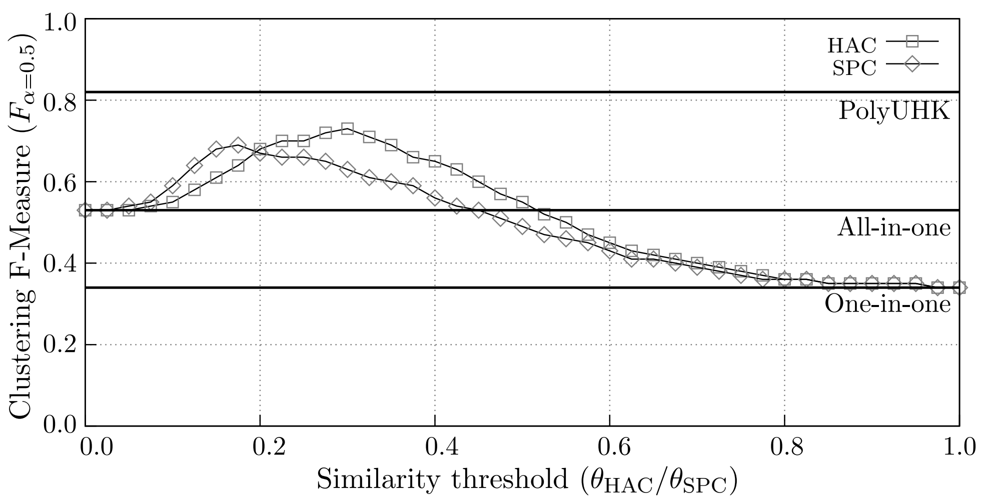


**Figure 4.4:** F-Measure of the unconstrained single pass clusterer (SPC) and hierarchical agglomerative clusterer (HAC) for all steps of the similarity threshold. The scores of the all-in-one and one-in-one baselines as well as of the best performing team (PolyUHK, [Chen et al., 2009]) are shown for comparison.

selected. In order to avoid confusion, it should be noted that the PolyUHK team was the best team in both the attribute extraction and the web page clustering task of the WePS-2 workshop.

### Constrained Clustering Performance

It is intuitive that the utility of the two constraint types depends on the similarity threshold of the clustering algorithm. If a threshold very close to 0 is used, most web pages will be contained in few big clusters. Therefore, must-link constraints are less likely to have an effect on the clustering. On the other hand, if the web pages are contained in a singleton cluster (similarity threshold above 1), then the clustering is not changed if cannot-link constraints are enforced. Thus, if the single constraint sets are enforced on the clustering algorithms, cannot-link constraints have a greater effect if a threshold close to 0 is chosen. The effect of must-link constraints increases the closer the threshold is to 1.

The effect of four single constraint sets on the performance of the single pass clusterer are shown in Figure 4.5. For all steps of similarity thresholds, the constraint strength threshold is determined by performing leave-one-out cross-validation as described in Section 4.1.3. As it was reasoned, cannot-link constraints affect the clustering for thresholds near to 0, while must-link constraints affect the clustering for higher thresholds. The results of the enforcement of the constraint sets on the hierarchical clusterer are very similar to those of the enforcement on the single pass clusterer.

As it was shown in Section 4.2.2, the constraint set generated from the school attribute generally benefits from soft matching. This can also be seen in the first graph of Figure 4.5. In the case of cannot-link constraints generated from the occupation attribute, however, the constraints have also a negative effect on clustering quality for some similarity thresholds. Since the constraint strength threshold was selected for each problem by cross-validation, the generated constraint sets have to perform very differently for the different problems. This is because, if the generated constraints have a negative effect for all problems, a constraint threshold of 1 is selected in cross-validation and therefore no constraints are employed in the actual clustering.

Although the constraint set of must-link constraints generated from the nationality attribute achieves the highest possible precision, it has only a small effect on clustering performance. Furthermore, the set of must-link constraints which is generated from the occupation attribute has a more positive effect on the clustering, although its precision is (in the case of the maximum method) considerable low (0.790 in absolute count). Therefore, the relatively high recall of the occupation constraint set (0.097 in absolute count) is able to outweigh the low precision of the constraint set. This is, however, only the case for similarity thresholds which are higher than the one for which the best performance is achieved.

The constraint sets can be separated based on their effect on the clustering algorithm. The first group is the group of constraint sets with a high precision, but low recall. These constraints have only a small effect on the clustering, but generally a positive one. For cannot-link constraints the sets generated from date of birth, email, school and third name are part of this first group. Must-link constraint sets which have this effect are those generated from date of birth, birthplace, email, nationality and school. the second group consists of constraint sets with a high recall, but low precision. These are the sets generated from affiliation and occupation for cannot-link constraints, and the one generated from occupation for must-link constraints. An exceptional constraint set in this consideration is the one of must-link constraints generated from the affiliation attribute. Nevertheless, although it has a high precision (0.956) and a high recall (0.060, both in absolute count), the F-Measure can only be marginally improved by its employment; in the case of the single pass clusterer, the highest achieved F-Measure score increases from 0.69 to 0.71 (multiplication method and soft matching).

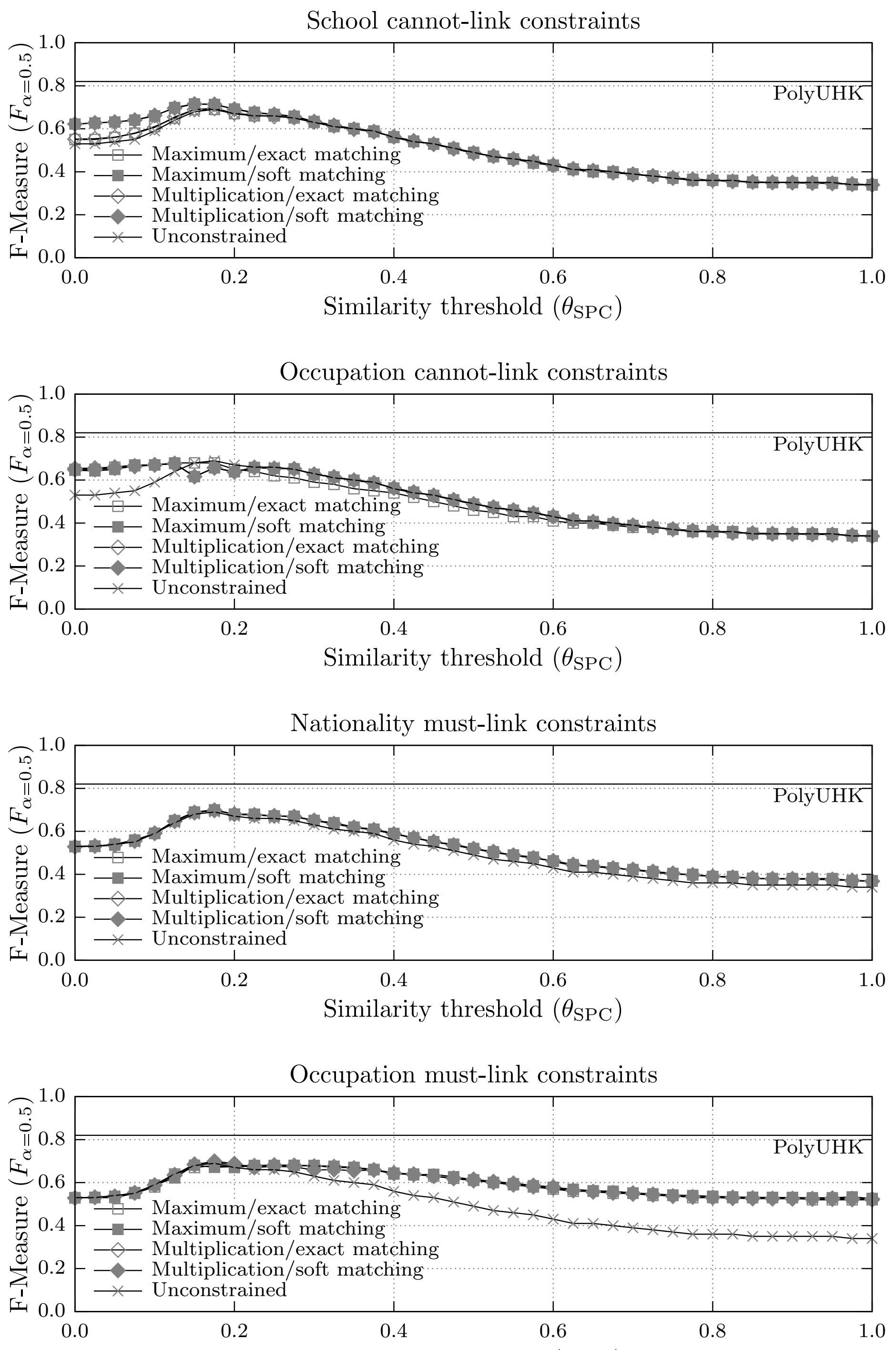


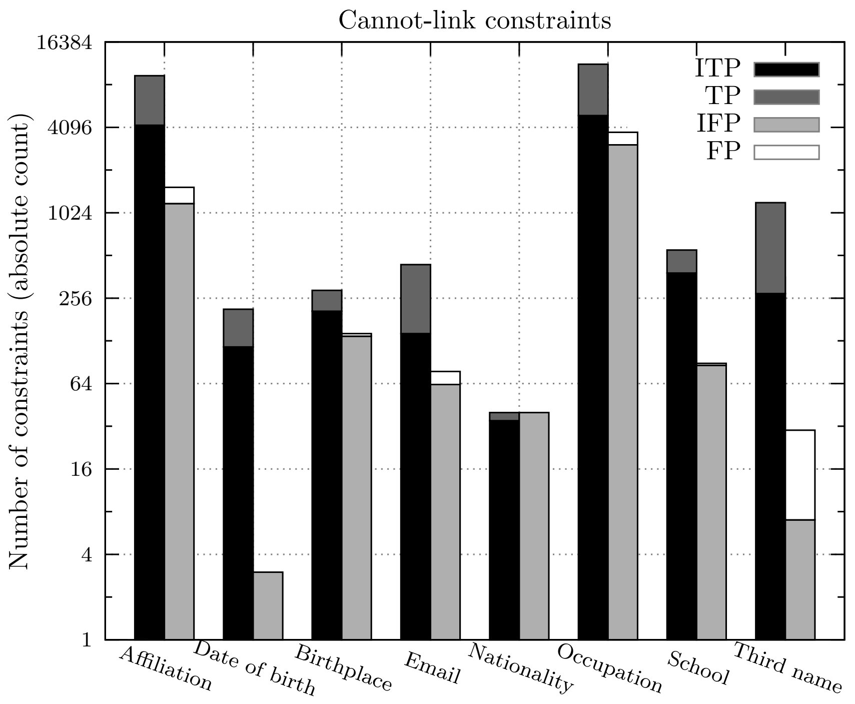
**Figure 4.5:** BCubed F-Measure achieved by the constrained single pass clusterer on employment of four different constraint sets. The performance of the unconstrained single pass clusterer and of the PolyUHK system are provided as reference points.

### Informativeness

Although, as shown in Figure 4.5, clustering performance increases for some similarity thresholds if constraint sets are employed, the top clustering performance does so only marginally. It can be assumed that this phenomenon is due to a low informativeness of the constraint sets. Therefore, if many of the generated constraints are already satisfied by the unconstrained clustering algorithm, the effect of the constraint set on the clustering is small. Empirical evidence for this assumption is shown in Figure 4.6. For each attribute, two columns are shown. The number of correct (left) and incorrect (right) constraints which are generated from the attribute values are represented by the height of the columns. The maximum method and exact matching was applied for constraint set generation. Since the scale of the ordinate is logarithmic, the precision of the generated constraints is expressed by the height difference of the two columns. Additionally, the number of constraints which are informative with respect to the single pass clusterer are shown as the lower part of the columns. Therefore, informative precision is expressed by the height difference of the lower parts of the two columns of one person attribute. Moreover, informativeness (either for correct or incorrect constraints) is inversely expressed by the height of the upper parts of the columns; the informativeness of the constraint sets generated from the different person attributes can be compared by comparing the size of the upper parts. Thus, with the exception of the constraints generated from the third name attribute, the must-link constraints tend to be less informative than the cannot-link constraints in both correct and incorrect constraints. Therefore, as it was assumed, a great part of the attribute information which suggests a grouping of web pages is redundant with respect to the clustering. However, the small effect of the cannot-link constraints on the clustering can not be explained by a low informativeness.

### Importance of Precision

Since incorrect constraints are likely to have a negative impact on the clustering, it is assumed that the precision of the constraints is of great importance for a high-quality constraint set. Nevertheless, it is shown in the prior experiments that recall is also of importance for a constraint set to be effective. In order to make a clearer statement on the relative importance of precision and recall for constraint set utility, the correlation between the metrics and the increase of clustering quality is measured. We employ the single problems of the corpus in conjunction with each of the single constraint sets for the calculation of the correlation. For each combination of problem, person attribute and constraint type, we compute the constraint set precision and recall (and informative precision/recall) as well as the change in F-Measure (with respect



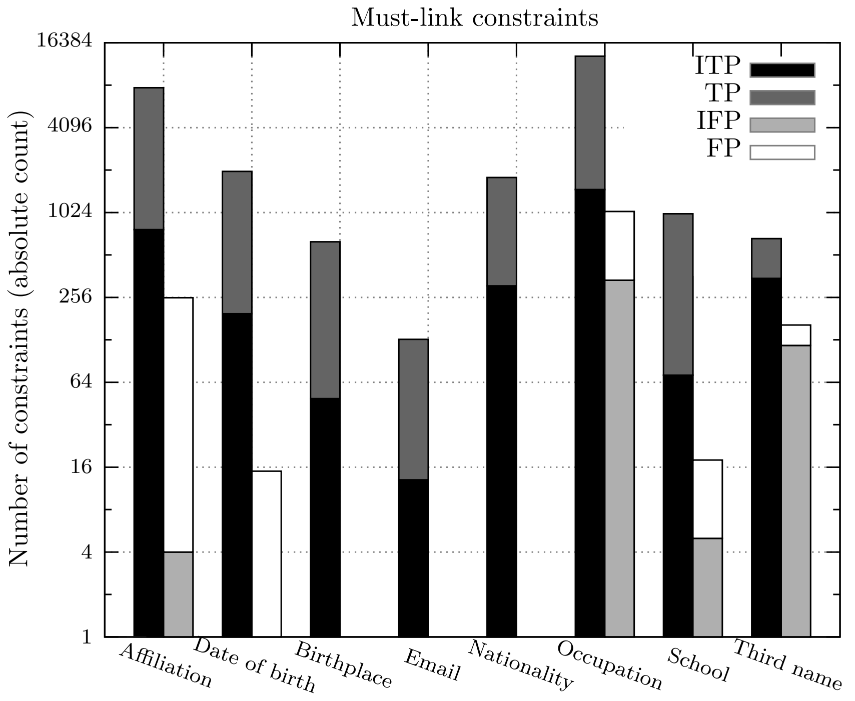


**Figure 4.6:** Number of correct (TP) and incorrect (FP) constraints generated by exact matching of the attribute values (logarithmic scale). The number of informative constraints (ITP, IFP) with respect to the single pass clusterer are included in the total number of constraints.

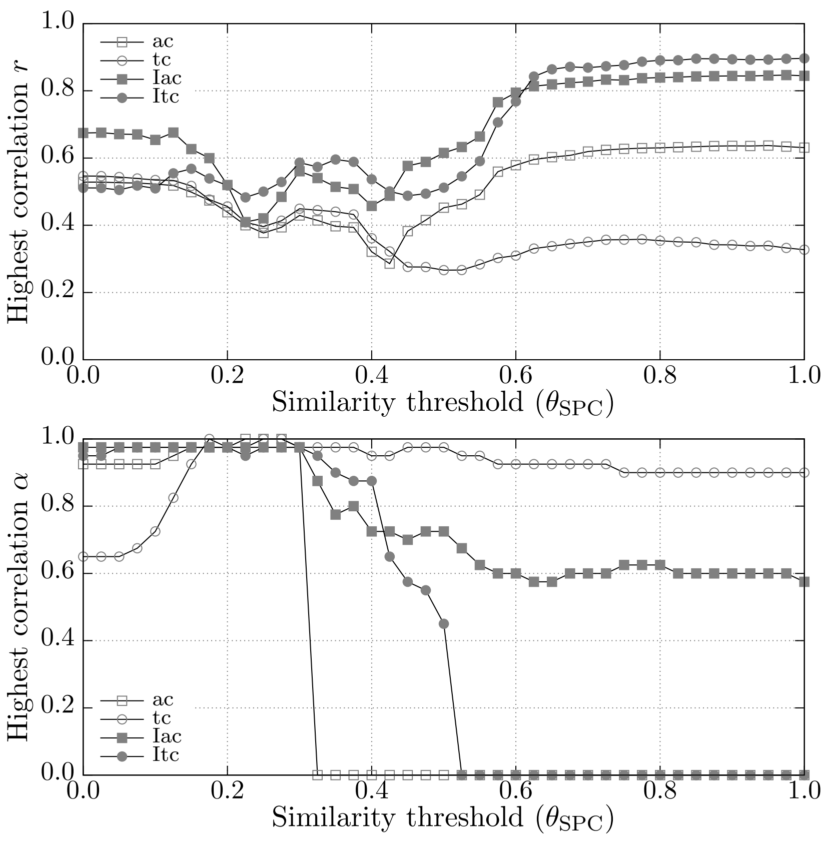


**Figure 4.7:** Best achievable correlation $r$ between constraint F-Measure and increase in clustering F-Measure ($\alpha$=0.5) for the single pass clusterer in dependence on the similarity threshold $\theta_{\text{SPC}}$ (top). The corresponding $\alpha$ of the constraint F-Measure is shown in the chart at the bottom. The constraint F-Measure is calculated using absolute (ac), transitive (tc), informative absolute (Iac) or informative transitive count (Itc).

to the unconstrained case) if the constraint set is applied. Precision and recall are then combined by the F-Measure. The correlation is calculated for different values of $\alpha$ and for all steps of similarity thresholds. Again, we employ the maximum method as well as exact matching and the single pass clusterer. Equivalent results are also obtained on the employment of the hierarchical clusterer. The highest achievable correlation for each threshold is shown in the upper part of Figure 4.7. The corresponding values of $\alpha$ for which the highest correlation is achieved are shown in the lower part of the figure.

As can be seen in the upper part of the figure, a generally higher correlation is achieved by utilizing informativeness. Therefore, the informativeness of the constraints is of importance for the effect of a constraint set on the clustering. Although there are differences in the correlation depending on if absolute or transitive count is applied, it remains unclear which method is better suited to measure the quality of a constraint set.

In all cases, however, the weakest correlation is found near the similarity threshold of 0.2. A possible reason for this phenomenon is that there are less informative constraints if the similarity threshold is chosen near this value. Consequently, the informative precision and recall have to be calculated from less constraints, thus loosing expressiveness. The "normal" precision and recall, however, are computed by considering mostly uninformative constraints.

The corresponding $\alpha$ for which the highest correlation was achieved is shown in the lower part of Figure 4.7. If $\alpha = 0$, then the highest correlation was achieved by only considering constraint recall. If $\alpha = 1$, only constraint precision was considered. For all but transitive count, precision becomes more important with a decreasing similarity threshold. We assume that this effect can be explained by the decreasing informativeness of must-link constraints and increasing informativeness of cannot-link constraints. Since the must-link constraints generally tend to be more precise (cf. Figure 4.6), it is less important to choose the constraint set with the highest precision.

### 4.2.4 Joint Application of Person Attributes

In order to find an answer to the questions under item 4, we employ constraint sets which consist only of must-link constraints, only of cannot-link constraints, or of a mixture of both constraint types. The effect on clustering quality is again measured by utilizing the BCubed F-Measure in comparison with the score of the unconstrained clustering algorithms. Then, the different methods we proposed for constraint strength addition and for conflict resolution are applied and compared by the resulting F-Measure score. Furthermore, we examine the effect of confidence weighting on the F-Measure score by employing the inverse referents-per-value and values-per-referent ratios as confidence factors for the corresponding constraints.

We see the highest potential for a positive effect in the combined application of both must-link constraints and cannot-link constraints. This is assumed since incorrect constraints of one constraint type can be discarded because of constraints of the other type in the fashion of a majority vote. However, it is further expected that this effect largely depends on appropriately calculated soft constraint strengths. We expect to achieve these appropriate constraint strengths through the application of (1) the multiplication method for constraint strength addition; (2) the person attribute confidence factors; (3) and the raising threshold conflict resolution.

### One Constraint-type

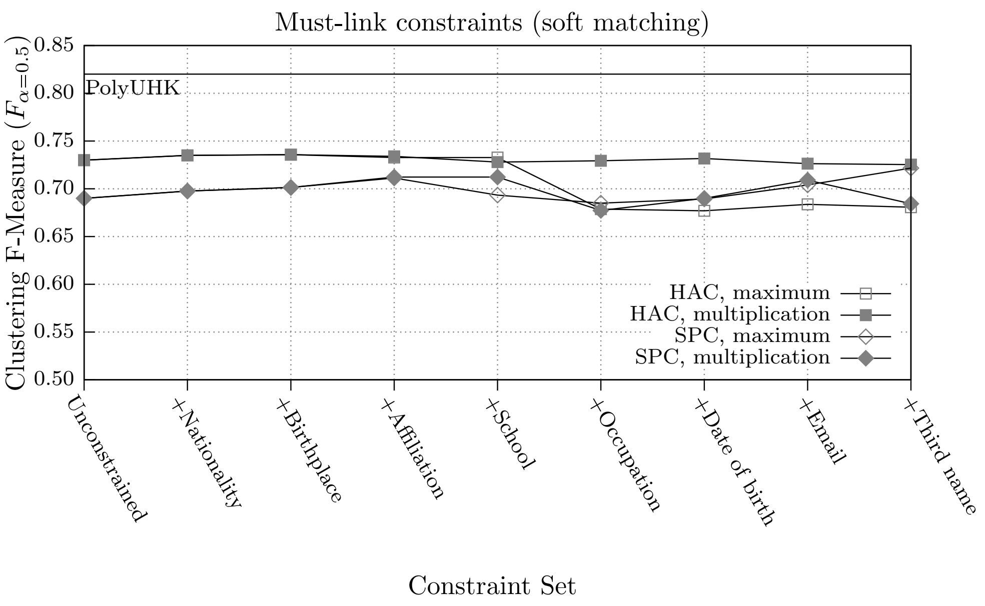
First, we examine the use of multiple person attributes, but only one constraint type. We sorted the single constraint sets by precision and added them iteratively. We apply soft matching of attribute values and either the maximum method or the multiplication method of constraint strength addition. The similarity threshold of the clustering algorithms and the constraint strength threshold are determined by leave-one-out cross-validation. The results are shown in Figure 4.8. As shown in the figure, only small improvements are achieved for both clustering algorithms. Moreover, especially in the case of cannot-link constraints, the constrained clustering algorithm performs often worse than the unconstrained variant.

The same experiment is also repeated for constraint sets which consists of both must-link and cannot-link constraints. We again sorted the single constraint sets by precision, but modified the order such that each person attribute is considered once before any attribute is considered a second time. Furthermore, since constraints of different types are employed within one constraint set, conflict resolution is applied in this experiment. The results are shown in Figure 4.9.

### Multiple Constraint-types

Other than on the application of constraints of only one type, the constrained clustering algorithms are able to consistently outperform the unconstrained variants. Even more, good increases in F-Measure are achieved by both algorithms. It should be noted that a first remarkable increase for both clustering algorithms is observed when the first set of cannot-link constraints—generated from the values of the third name attribute—is added. While the F-Measure of the hierarchical clusterer increases steadily with the addition of more constraint sets, this is not the case for the single pass cluster. Moreover, the performance of the single pass clusterer varies much more with the parameter setting than the performance of the hierarchical clusterer. Nevertheless, it remains unclear which method performs best, since some methods are able to benefit from constraint sets which mislead other methods. For both algorithms, however, the differences between the parameter settings increase with the number of considered person attributes. Therefore, selecting an appropriate method or strategy becomes more important the more constraint sets are incorporated.

We now inspect in more detail the constraint set which contains must-link as well as cannot-link constraints of all person attributes. To this end, the effect of the constraint set on the clustering algorithm is again shown in dependence on the similarity threshold. The constraint strength threshold is



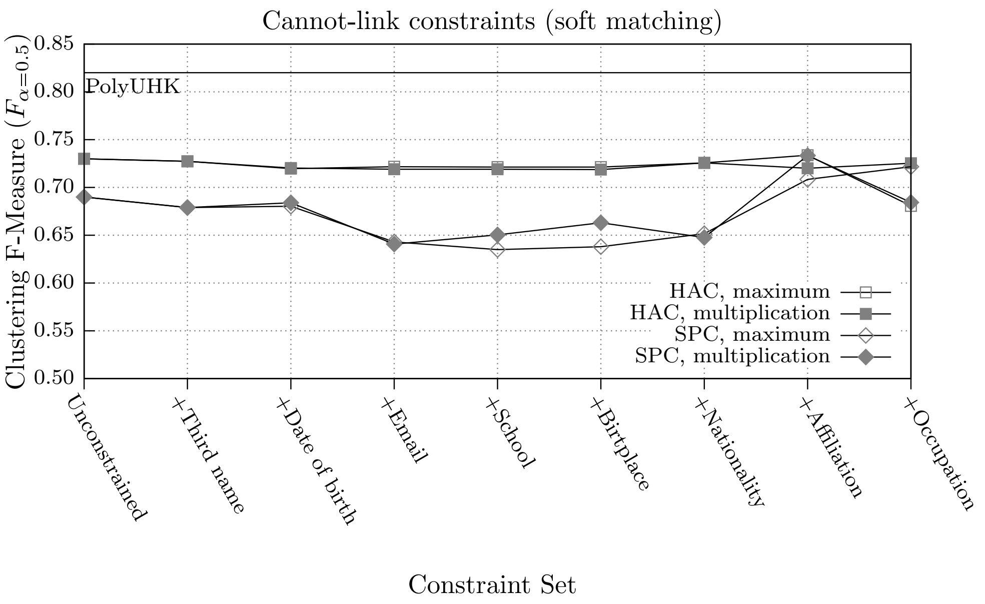
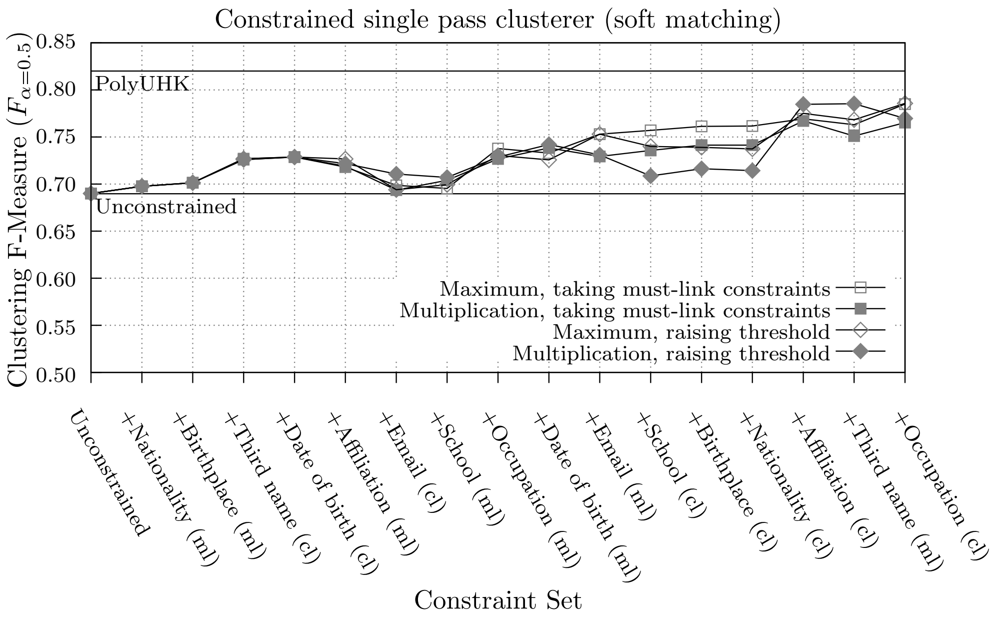


**Figure 4.8:** Clustering F-Measure for sets of must-link (top) or cannot-link constraints (bottom). Starting from an empty set of constraints ("Unconstrained"), the single constraint sets of all person attributes are added one at a time.



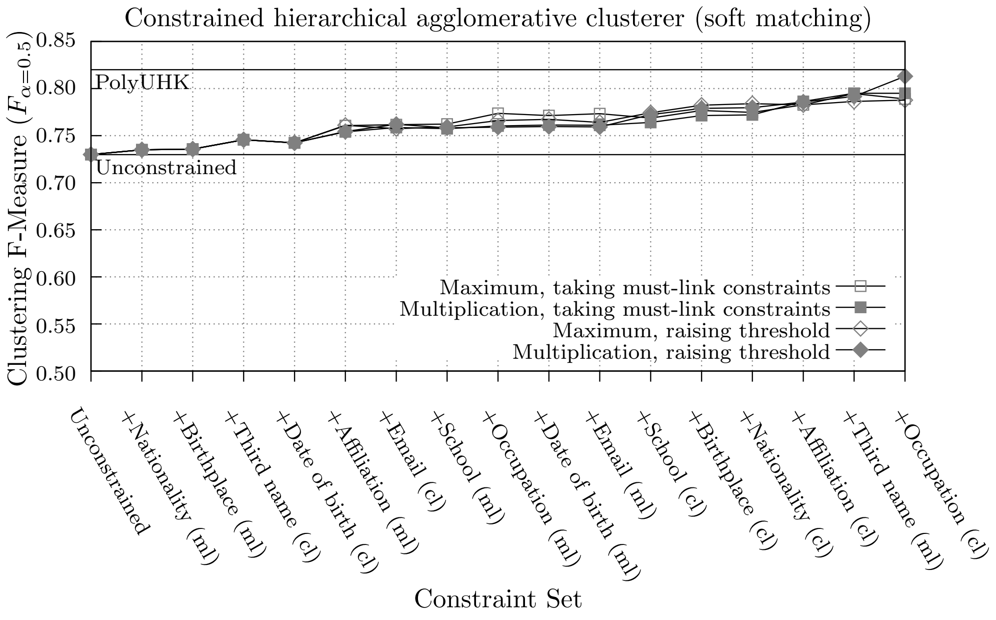
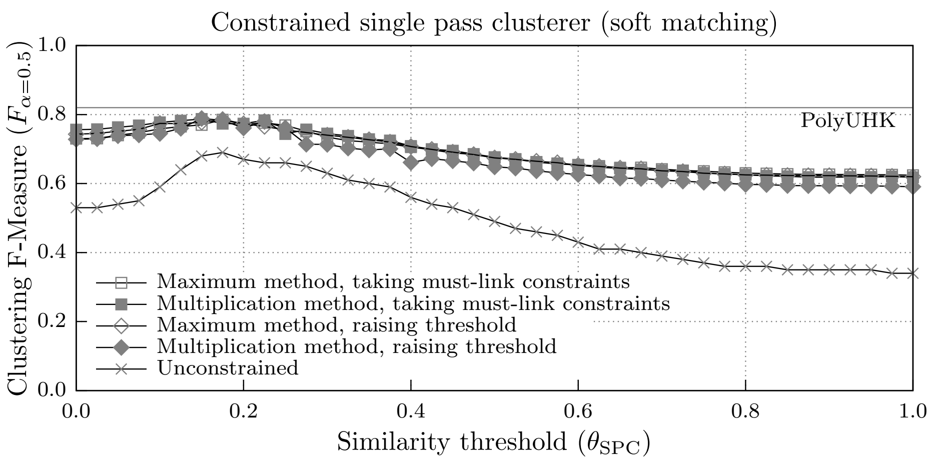


**Figure 4.9:** Clustering F-Measure for sets of must-link and cannot-link constraints. Similar to Figure 4.8, the single constraint sets of all person attributes are added to the set. Sets of must-link constraints are denoted by (ml), while sets of cannot-link constraints are denoted by (cl). The constraint set is imposed on the single pass clusterer (top) as well as on the hierarchical clusterer (bottom).



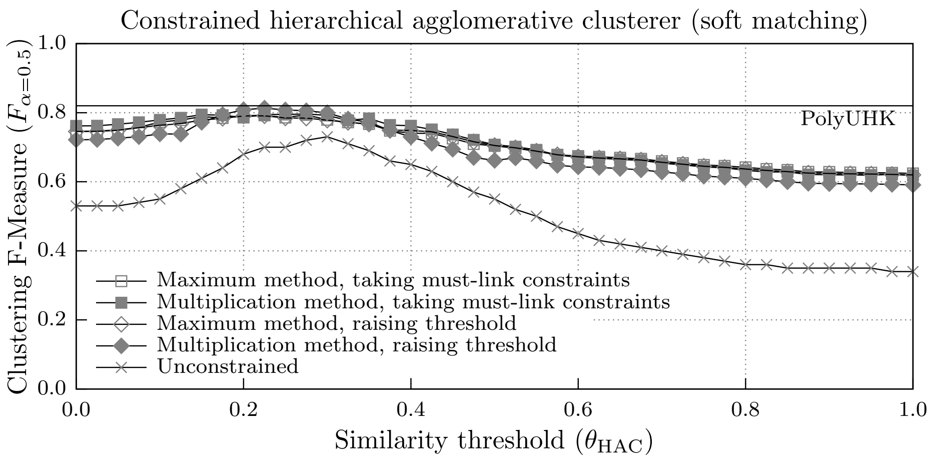


**Figure 4.10:** Clustering F-Measure of both constrained clustering algorithms on enforcement of the constraint set generated from all attribute values and constraint types. The score is shown in dependence on the similarity threshold. The F-Measure score of the best performing system of the WePS-2 workshop (PolyUHK) is shown for reference.

again determined by leave-one-out cross-validation for all steps of the similarity threshold. The result is shown in Figure 4.10. Although both constrained algorithms have very similar graphs, the constrained hierarchical clusterer can achieve a higher maximum F-Measure. In both cases, the combination of multiplication method and raising threshold conflict resolution has the lowest F-Measure of all combinations for most values of the similarity threshold. Nevertheless, near the maximum of the graphs this combination is on par with the other methods (single pass clusterer) or can even outperform them (hierarchical clusterer).

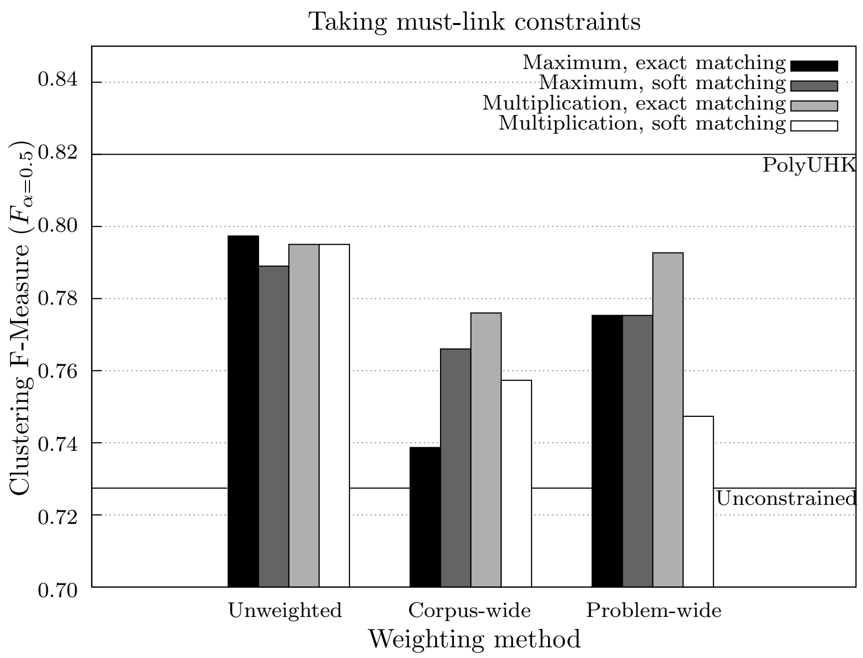
We will now consider the extreme cases of the similarity threshold. If the similarity threshold is 0, then the clustering algorithm merges all clusters which are not connected by a cannot-link constraint. The must-link constraints affect the clustering process by defining the initial situation at which the algorithm then starts. Since the maximum F-Measure score is not achieved at a threshold of 0, the algorithm is still able to find clusters which should be separated, but are not connected by a cannot-link, at higher thresholds. Moreover, the F-Measure achieved at a threshold of 0 is much higher in the case of the constrained clusterer ($\mathrm{P}_{\mathrm{eB}^3} = 0.70$, $\mathrm{R}_{\mathrm{eB}^3} = 0.85$) than for the unconstrained variant ($\mathrm{P}_{\mathrm{eB}^3} = 0.43$, $\mathrm{R}_{\mathrm{eB}^3} = 1.00$, values are calculated for exact matching, the maximum method and the raising threshold strategy). As it is shown by the values for BCubed precision and recall, a much higher precision is achieved at the cost of some recall.

If the similarity threshold is 1, then the clustering algorithm will only merge clusters which have identical web pages, but are not connected by a must-link or cannot-link constraint. While all must-link constraints are applied, cannot-links have only an effect in the situation of identical web pages. Therefore, the utility of the must-link constraints can be deducted from this situation; through the enforcement of the must-link constraints for a similarity threshold of 1, the BCubed precision decreases from 1.00 to 0.94, but the recall increases from 0.24 to 0.51 (values again calculated for exact matching, the maximum method and the raising threshold strategy). Nevertheless, the clustering algorithm is also able to improve on this situation. Therefore, the algorithm can successfully identify clusters which should be grouped, but are not connected by a must-link constraint.

### Constraint Weighting

Finally, we consider the weighting of the single constraint sets for their application within the complete constraint set. To this end, we again created a constraint set which includes constraints of all types and from all person attributes (we will refer to this as *unweighted*). Additionally, we created two further constraints sets in the same manner, but by additionally employing confidence factors. We arbitrarily decided to apply the inverse of the values-per-referent ratio as a confidence factor for cannot-link constraints, and the inverse of the referents-per-value ratio as a factor for must-link constraints. A different possibility would have been to employ the precision of the resulting constraint sets. We further distinguish between *corpus-wide* weighting by employing the average referents-per-value or values-pre-referent ratios ($\mu_{\mathrm{rpv}_{tn}}$,$\mu_{\mathrm{vpr}_{tn}}$) and *problem-wide* weighting by applying the referents-per-value or values-per-referent ratios of the current problem ($\mathrm{rpv}_{tn}$,$\mathrm{vpr}_{tn}$). The results are shown in Figure 4.11.

Quite counterintuitively, the F-Measure of the applied methods actually



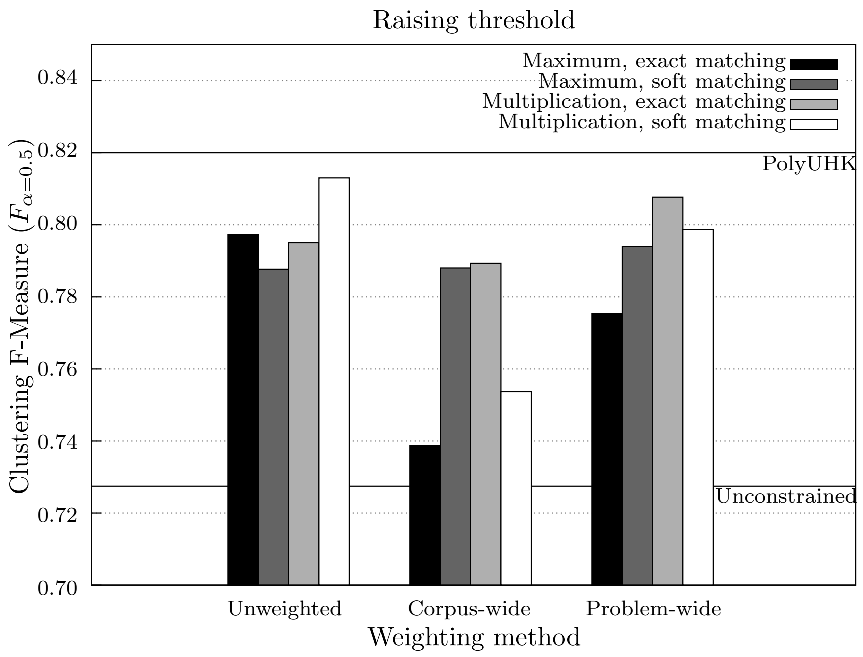


**Figure 4.11:** Clustering F-Measure of the constrained hierarchical clusterer on enforcement of the constraint set generated from all attribute values and constraint types. The score is shown for different methods of constraint strength addition and attribute value matching (shading), for different weighting methods (unweighted, corpus-wide or problem-wide) and for different conflict resolution strategies (top/bottom). The score of the PolyUHK system as well as the unconstrained hierarchical clusterer are shown as reference.

decreases in most cases when weighting is applied. An increase in F-Measure score is only achieved for problem-wide weighting and if the raising threshold strategy is utilized. In this case, especially the multiplication method in conjunction with exact matching is able to benefit from the weighting. Since the introduction of confidence factors was partially motivated with this combination in mind, it might be necessary to prove if the application is also justified in the case of the other combinations. Still, the effect of confidence weighting on the other methods is quite considerable. This shows the importance of appropriate weighting, even if the selected weights have actually a negative effect in this case.

### 4.2.5 Algorithmic Attribute Extraction

We address the question under item 5 by comparing the person attribute values of the gold standard of the WePS-2 workshop with the person attribute values which are extracted by the attribute extraction system which performed best in the workshop. This comparison is based on (1) the constraint set precision and recall of the constraint sets containing constraints of one type which are generated from the values of one person attribute; (2) and the BCubed F-Measure achieved through the application of the (either weighted or unweighted) set of constraints generated from all person attribute values. It is assumed that the algorithmically extracted person attribute values result in a lower constraint set precision, constraint set recall and clustering F-Measure.

#### Precision and Recall

In order to compare the extracted person attribute values, we again start by considering the single constraint sets like in Section 4.2.1. We repeat exactly the same experiment (exact matching, maximum method, one constraint type and person attribute per constraint set) with the person attribute values extracted by the PolyUHK attribute extraction system. The results are shown together with those achieved with the gold standard attribute values in Table 4.3.

In the case of cannot-link constraints, a lower precision and a lower or equal recall is achieved through the utilization of the PolyUHK person attribute values. It should be noted that for the nationality attribute no cannot-link constraint is generated, and therefore no precision exists for this attribute. This is simply because no value for nationality is extracted by the system. In the case of the third name attribute, however, an considerably high recall is reached. Nevertheless, the precision is rather low. Overall the highest precision of a cannot-link constraint set is accomplished for the email attribute, but with

**Table 4.3:** Precision (P) and recall (R) of constraint sets of one type and generated from the values of one person attribute either by employing the values of the gold standard (gold) or those extracted by the PolyUHK attribute extraction system (UHK). All values are calculated by absolute count.

| | Cannot-link constraints | | | | Must-link constraints | | | |
|---|---|---|---|---|---|---|---|---|
| Attribute | $P_{gold}$ | $P_{UHK}$ | $R_{gold}$ | $R_{UHK}$ | $P_{gold}$ | $P_{UHK}$ | $R_{gold}$ | $R_{UHK}$ |
| Affiliation | 0.747 | 0.532 | 0.092 | 0.006 | 0.956 | **1.000** | 0.060 | 0.001 |
| Date of birth | 0.989 | 0.624 | 0.002 | 0.006 | 0.933 | **1.000** | 0.015 | 0.007 |
| Birthplace | 0.773 | 0.599 | 0.003 | 0.004 | **1.000** | **1.000** | 0.005 | 0.002 |
| Email | 0.945 | **0.873** | 0.004 | $\sim 0.0$ | 0.929 | **1.000** | 0.001 | $\sim 0.0$ |
| Nationality | 0.750 | - | $\sim 0.0$ | 0.000 | **1.000** | - | 0.014 | 0.000 |
| Occupation | 0.658 | 0.528 | **0.115** | 0.009 | 0.790 | 0.911 | **0.097** | 0.004 |
| School | 0.870 | 0.659 | 0.005 | $\sim 0.0$ | 0.903 | **1.000** | 0.008 | $\sim 0.0$ |
| Third name | **0.996** | 0.546 | 0.016 | **0.039** | 0.743 | 0.784 | 0.007 | **0.023** |

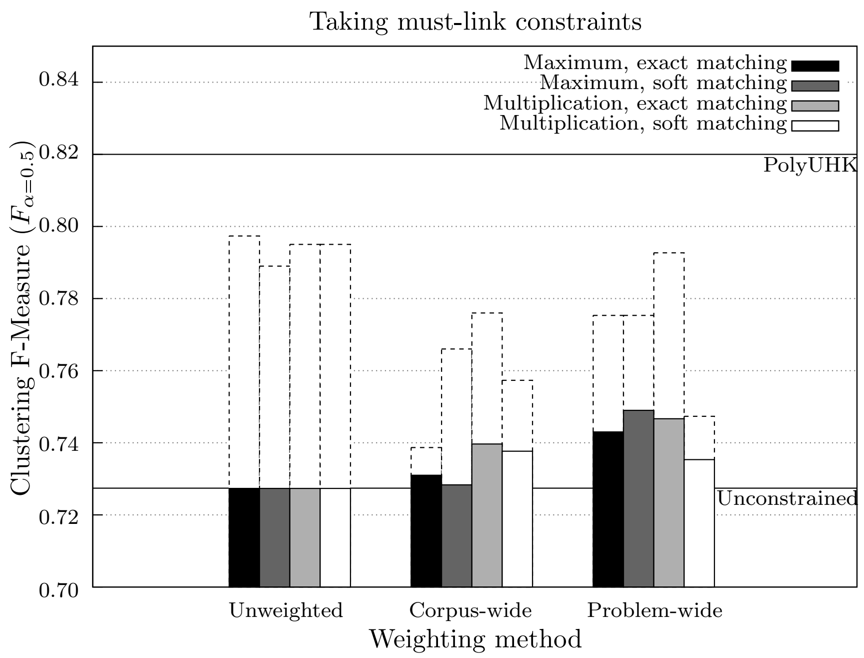
a very low recall.

On the other hand, the precision of the generated must-link constraint sets is for most constraint sets very high. But also in theses cases the recall is lower than for the gold standard person attribute values. There are, however, two exceptions to this; (1) no must-link constraints are generated by considering the nationality person attribute; (2) the third name attribute has a lower precision (still higher than the precision for the gold standard attribute values) but also a higher recall. Therefore, the algorithmically extracted values for the third name attribute are likely to be more suited for constraint generation than the original ones.

### Confidence Factors

Finally, we want to test the suitability of our framework for algorithmically extracted person attribute values. To this end, we repeat the last experiment of Section 4.2.4, but by employing the automatically extracted person attribute values instead of the gold standard. Furthermore, the confidence factors (inverse referents-per-value and values-per-referent ratios) are adjusted to the algorithmically extracted person attribute values. The result is shown in Figure 4.12, with the results from the employment of the gold standard attributes represented by empty bars.

Although the effect of confidence weighting was mostly negative in the case of gold standard attributes, it is mostly positive in the case of algorithmic attribute values. Moreover, although the constraint generation has no effect on



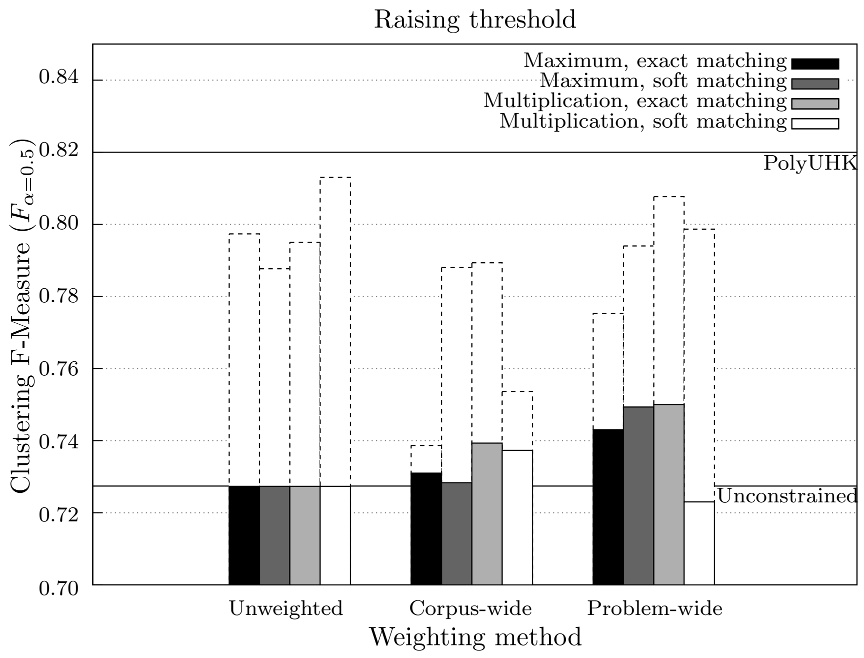


**Figure 4.12:** Clustering F-Measure of the constrained hierarchical clusterer on enforcement of the constraint set generated from all attribute values extracted by the PolyUHK attribute extraction system. The achieved F-Measure for the same parameters and by employing the gold standard attributes (cf. Figure 4.11) are shown as empty bars. The score of the PolyUHK clustering system as well as the unconstrained hierarchical clusterer are shown for reference.

clustering quality if the single constraint sets are not weighted, the clustering quality can be increased if confidence weighting is employed. Even more, problem-wide confidence weighting can again improve on corpus-wide weighting. This is, however, not the case for the combination of soft matching and the multiplication method. We therefore assume that different weights are needed for this combination of methods. Furthermore, although the F-Measure for algorithmically extracted attribute values is considerably lower than the F-Measure for gold standard values, we assume that it is possible to approach this score by more sophisticated methods for confidence weighting.

# Chapter 5

# Conclusions

In this thesis, we proposed a framework for the generation of hard instance-based constraints from person attribute values. Within this framework, we introduced two methods for the addition of constraint strengths and two strategies for the resolution of conflicts within constraint sets. Furthermore, we proposed two statistical properties of person attributes and showed that they are indicators for the suitability of a person attribute for constraint generation. Moreover, we showed that the soft matching of person attribute values is beneficial for some of the person attributes, but disadvantageous for others. We demonstrated that constraints are suited to guide clustering algorithms in web people search by the enforcement of constraint sets on two clustering algorithms and the employment of the WePS-2 corpus. In this context, we showed that constraint set precision is more important than recall and that the prediction of the effect of constraint sets on clustering quality is improved when the informativeness of the constraints is considered. Furthermore, we gave empirical evidence that a combination of must-link and cannot-link constraints is able to outperform constraint sets of only one constraint type. Moreover, it was shown that clustering quality is affected by the weighting of constraints via person attribute confidence factors, but also that an intuitive solution can have a disadvantageous effect. However, none of our proposed methods for constraint strength addition or conflict resolution was able to consequently outperform the other. Finally, it was shown that the framework can be adjusted to less complete and less accurate person attribute values by the adjustment of the person attribute confidence factors.

Nevertheless, also new questions arise from the results of our experiments. Since the benefit of performing soft matching of person attribute values depends much on the person attribute, it might be advantageous to introduce for each person attribute a similarity threshold for soft matches. Then only soft matches with a matching score above or equal to the threshold are considered. Such an approach is even more beneficial, if it is possible to develop a method for automatically determining this threshold. To this end it might be possible to create a measure for constraint set quality which does not require the gold standard, for example based on constraint set coherence [Davidson et al., 2006]. Alternatively, a relational approach as it is used by Bhattacharya and Getoor [2007] might also be applicable. Then, an appropriate threshold might be determinable by analyzing the change of this measure for different similarity thresholds (cf. [Salvador and Chan, 2004]). Moreover, it might also be possible to apply such a method for the selection of appropriate confidence factors for the different person attributes and constraint types.

We expect that the importance of conflict resolution increases together with the quality of the applied confidence factors. Therefore strategies for conflict resolution should be developed which also account for situations in which multiple conflicts are resolvable by the removal of a single constraint. Alternatively, the application of a constrained clustering algorithm which employs soft constraints can be considered. In this case, no conflict resolution has to be applied since the clustering algorithm is not required to satisfy all constraints. In this context it might also be advantageous to consider metric-learning algorithms such as those by Bilenko et al. [2004] and Bar-Hillel et al. [2005].

Finally, although the framework for constraint generation was developed for web people search, it can also be used for other tasks. In general it can be applied to all tasks in which domain specific knowledge exists but is not completely reliable.

# Appendices

# Appendix A

# BCubed Metrics for Overlapping Clusters

To understand how the BCubed metrics are extended for overlapping clusters, it is helpful to rewrite the Equations 2.4 of page 11 as:

$$\mathrm{p}_{\mathrm{B}^3}(d) = \frac{1}{|\{d' \in D : C(d') = C(d)\}|} \sum_{d' \in D : C(d') = C(d)} \delta_{r(d), r(d')}$$

$$\mathrm{r}_{\mathrm{B}^3}(d) = \frac{1}{|\{d' \in D : r(d') = r(d)\}|} \sum_{d' \in D : r(d') = r(d)} \delta_{C(d), C(d')}$$

with $\delta_{a,b}$ being

$$\delta_{a,b} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases}$$

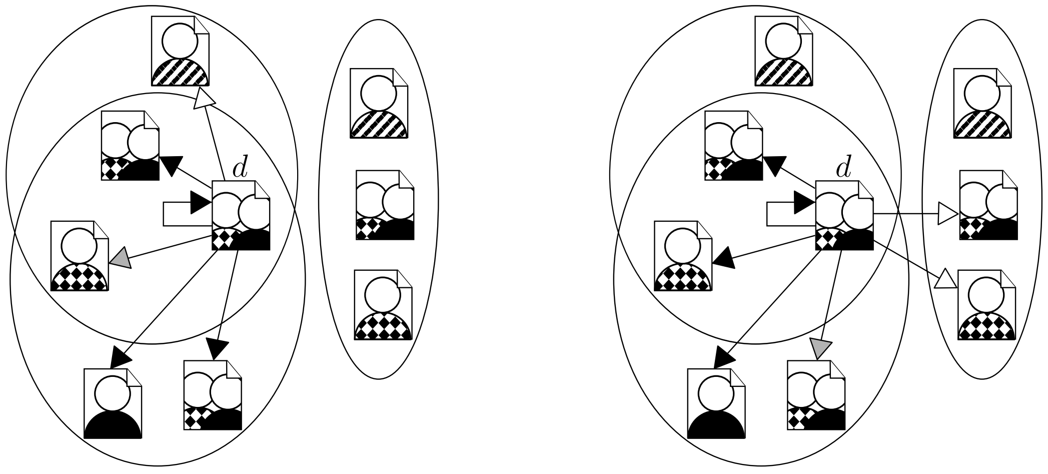
Then, $C(d)$ is replaced by $\mathcal{C}(d)$, which denotes the set of clusters to which $d$ is assigned to. Similarly, $r(d)$ is replaced by the set of referents which $d$ concerns, $R(d)$. Thus, single items are replaced by sets. Therefore, instead of checking for equality it is now checked if the two sets have at least one item in common:

$$\mathrm{p}_{\mathrm{eB}^3}(d) = \frac{1}{|\{d' \in D : \mathcal{C}(d') \cap \mathcal{C}(d) \neq \{\,\}\}|} \cdot \sum_{d' \in D : \mathcal{C}(d') \cap \mathcal{C}(d) \neq \{\,\}} \mathrm{p_m}(d, d')$$

$$\mathrm{r}_{\mathrm{eB}^3}(d) = \frac{1}{|\{d' \in D : R(d') \cap R(d) \neq \{\,\}\}|} \cdot \sum_{d' \in D : R(d') \cap R(d) \neq \{\,\}} \mathrm{r_m}(d, d')$$

with $\mathrm{p_m}$ and $\mathrm{r_m}$ being replacements for $\delta$.

Since multiple-referent pages and overlapping clusters are possible, the binary $\delta$-function is no longer sufficient. Consider, for example, an algorithm



$\mathrm{p_{eB^3}}(d) = \frac{4.5}{6}$ $\qquad$ $\mathrm{r_{eB^3}}(d) = \frac{4.5}{7}$

**Figure A.1:** Example calculation of extended BCubed precision (left) and recall (right). Multiple-referent pages (here only dual-referent pages) show multiple figures. Each arrow symbolises one calculation of $\mathrm{p_m}$ (left) or $\mathrm{r_m}$ (right) for the web page $d$. The shading of the arrow's head represents the result: 0 (white), 0.5 (gray) or 1 (black).

which correctly clustered a web page $d$ with other pages concerning one referent of $R(d)$. The other referents of $R(d)$, however, were not taken into account by the algorithm. Therefore, a score between 1 (full success) and 0 (complete failure) should be given. For the extended BCubed metrics, these scores are calculated as:

$$\mathrm{p_m}(d, d') = \frac{\min(|\mathcal{C}(d) \cap \mathcal{C}(d')|, |R(d) \cap R(d')|)}{|\mathcal{C}(d) \cap \mathcal{C}(d')|}$$

$$\mathrm{r_m}(d, d') = \frac{\min(|\mathcal{C}(d) \cap \mathcal{C}(d')|, |R(d) \cap R(d')|)}{|R(d) \cap R(d')|}$$

Thus, if there are less common clusters than common referents, $\mathrm{r_m}$ is low. On the other hand, $\mathrm{p_m}$ is low, if the number of common clusters is higher than the number of common referents. An example of the calculation is shown in Figure A.1.

Final precision and recall are then calculated like for the standard BCubed metrics:

$$\mathrm{P_{eB^3}} = \frac{1}{|D|} \cdot \sum_{d \in D} \mathrm{p_{eB^3}}(d) \qquad \mathrm{R_{eB^3}} = \frac{1}{|D|} \cdot \sum_{d \in D} \mathrm{r_{eB^3}}(d)$$

# Appendix B

# String Similarity Metrics

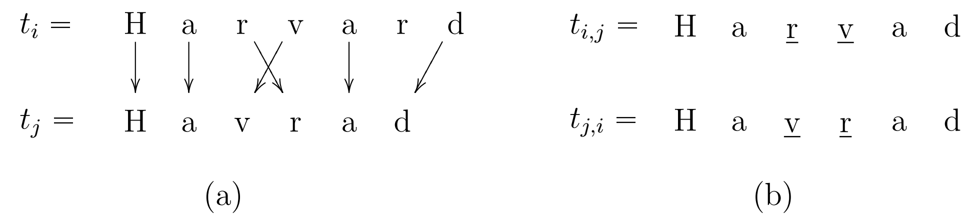
The soft-tf·idf character-string similarity-metric calculates a similarity score for two strings based on a collection of such strings [Cohen et al., 2003]. While soft-tf·idf itself is applied on the whole character string, it employs a second similarity-metric for word-wise similarity-computation. Like in the paper by Cohen et al., we utilize the Jaro-Winkler metric[1] [Winkler, 1990] as secondary metric, which is a modification of the Jaro metric. Section B.1 introduces the Jaro metric. The Jaro metric is then extended to the Jaro-Winkler metric in Section B.2. Finally, the soft-tf·idf metric is detailed in Section B.3.

## B.1 Jaro metric

The Jaro metric computes the similarity of two character-strings using the number of common characters and transpositions in the strings [Winkler, 1990]. To compute the common characters of $t_i$ and $t_j$, the characters of one of the two strings—$t_i$ is used in this explanation—are considered one at a time. Each character is *aligned* with the first identical character of $t_j$ within a certain search range. No character in $t_j$ can be aligned to multiple characters of $t_i$. The search range is given by $\max(|t_i|, |t_j|)/2 - 1$, where $|t_i|$ is the number of characters in $t_i$. Note that, while both strings have the same characters in common with each other, the positions of the common characters in the two strings may be different. The common characters of $t_i$ and $t_j$ in the order in which they appear in $t_i$ are denoted by $t_{i,j}$. The number of transpositions, $\tau$, is then defined as:

$$\tau(t_i, t_j) = \frac{1}{2} \sum_{k=1}^{|t_{i,j}|} (1 - \delta_{t_{i,j}(k), t_{j,i}(k)})$$

[1]Precisely, we employ the class JaroWinklerDistance of the *LingPipe* Java library, which can be accessed on `http://alias-i.com/lingpipe`.



$$\varphi_{\text{j}}\big(\text{“Harvard”}, \text{“Havrad”}\big) = \frac{1}{3} \cdot \left( \frac{\big|\text{“Harvad”}\big|}{\big|\text{“Harvard”}\big|} + \frac{\big|\text{“Harvad”}\big|}{\big|\text{“Havrad”}\big|} + \frac{\big|\text{“Harvad”}\big| - \frac{2}{2}}{\big|\text{“Harvad”}\big|} \right)$$
$$= 0.897$$

**Figure B.1:** Example of the calculation of the Jaro metric for the character strings “Harvard” and “Havrad”: (a) alignment of characters; (b) half-transpositions; (bottom) final calculation and result.

where $t_{i,j}(k)$ is the character at position $k$ in $t_{i,j}$, and $\delta_{t_{i,j}(k),t_{j,i}(k)}$ equals to 1 if $t_{i,j}(k) = t_{j,i}(k)$ and 0 otherwise. The final Jaro metric is then defined as:[2]

$$\varphi_{\text{j}}(t_i, t_j) = \frac{1}{3} \cdot \left( \frac{|t_{i,j}|}{|t_i|} + \frac{|t_{i,j}|}{|t_j|} + \frac{|t_{i,j}| - \tau(t_i, t_j)}{|t_{i,j}|} \right)$$

The Jaro metric weighs mismatches in characters over transpositions, which are a frequent type of writing errors. An example for the calculation of the Jaro metric for a term with typographical errors is provided in Figure B.1.

## B.2 Winkler Modification of the Jaro Metric

The modification by Winkler accounts for the increase of the frequency of typographical errors with increasing character position in a word [Winkler, 2006]. Differences at the start of words are therefore less likely to be caused by writing mistakes. If the two character strings which are compared have a common prefix, the achieved similarity-score of the Jaro metric is increased by the modification of Winkler. Let $p_{t,t'}$ be the greatest common prefix of the two strings $t$ and $t'$. The Jaro-Winkler metric is then defined as:

$$\varphi_{\text{jw}}(t, t') = \begin{cases} \varphi_{\text{j}}(t, t') & \text{if } \varphi_{\text{j}}(t, t') < \theta_{jw} \\ \varphi_{\text{j}}(t, t') + 0.1 \cdot \min(|p_{t,t'}|, 4) \cdot (1 - \varphi_{\text{j}}(t, t')) & \text{if } \varphi_{\text{j}}(t, t') \geqslant \theta_{jw} \end{cases}$$

[2]The original metric, as it is described by Winkler, employed three weights for the three different fractions of the equation. The *LingPipe* implementation, however, uses the fixed value of 1/3 for each of them, as it is described in the equation.

That is, the score is only increased if the Jaro metric would have reached the threshold $\theta_{jw}$. We use $\theta_{jw} = 0.7$ in our experiments, as it is suggested in the documentation of *LingPipe*[3].

In the example of Figure B.1, the first two characters of the two strings are identical. Thus, the similarity will be increased by $0.2 \cdot (1 - 0.897)$. Hence, the Jaro Winkler similarity of the two strings is 0.918.

## B.3 Soft-tf·idf Metric

The Jaro-Winkler metric was designed for the comparison of person names within the U.S. Census [Winkler, 2006]. Person attribute values, however, range from initials to complete company names. Therefore, we employ the soft-tf·idf string-similarity metric, which is proposed by Cohen et al. [2003], for the whole person attribute values. This metric then applies the Jaro-Winkler metric on all pairs of words of the different person attribute values. The final score is then computed using the word-wise scores. Like in web page representation, we remove all words from the person attribute values which have a noncrucial affect on the meaning (e.g., "in", "for", "the" and "of"). Thus, person attribute values match even if these terms are omitted from one of them.

The soft-tf·idf metric employs the tf·idf formula (cf. Equation 2.1 on page 7) to determine the weight of each term $t$ of the complete person attribute value $v$. In this context, a term is a single word within a person attribute value. The person attribute value is then seen as a document. Instead of the collection of web pages, all person attribute values of the same person attribute $a$, $V_a$, are used. The following weighting formula is then employed within the soft-tf·idf metric:

$$w_s(t, v) = \frac{\text{tf·idf}(t, v)}{\sqrt{\sum_{t' \in v} \text{tf·idf}(t', v)^2}}$$

which is then applied to compute the similarity $\phi_s$ of the two terms $t$ and $t'$, with $t \in v$ and $t' \in v'$ respectively:

$$\phi_s(t, v, t', v') = \begin{cases} 0 & \text{if } \varphi_{\text{jw}}(t, t') < \theta_s \\ w_s(t, v) \cdot w_s(t', v') \cdot \varphi_{\text{jw}}(t, t') & \text{if } \varphi_{\text{jw}}(t, t') \geqslant \theta_s \end{cases}$$

with $\theta_s$ being a threshold on the Jaro-Winkler metric for counting the two terms as a match. Like in the original publication, we set the threshold to 0.9 [Cohen et al., 2003]. The complete similarity, $\varphi_s$, is then computed by summing up the $\phi_s$-similarities between each term $t$ in the first attribute value $v$, and the

[3] `http://alias-i.com/lingpipe`

most similar ("closest") term to $t$ in the second value $v'$. More formally, the most similar term is defined as:

$$\text{closest}(t, v') = \underset{t' \in v'}{\arg\max}\ \varphi_{\text{jw}}(t, t')$$

which is then used in the complete formula:

$$\varphi_s(v, v') = \sum_{t \in v} \phi_s(t, v, \text{closest}(t, v'), v')$$

An example for the calculation of $\varphi_s$ is shown in Figure B.2.

As can be seen, $\sqrt{\sum_{t' \in v} \text{tf·idf}(t', v)^2}$ is a normalization factor like in the cosine similarity. Without the multiplication by the Jaro-Winkler similarity, which does not change the bounds of the whole metric as it is always in the range of $[0, 1]$, the equation can be written as:

$$\varphi_{s'}(v_i, v_j) = \sum_{t_i \in v_i} \frac{\text{tf·idf}(t_i, v_i)}{\sqrt{\sum_{t'_i \in v_i} \text{tf·idf}(t'_i, v_i)^2}} \cdot \frac{\text{tf·idf}(\text{closest}(t_i, v_j), v_j)}{\sqrt{\sum_{t'_j \in v_j} \text{tf·idf}(t'_j, v_j)^2}}$$

which is similar to the cosine similarity (cf. Equation 2.2 on page 7). Since, terms of a set are used instead of vectors, the dot-product and the norms are replaced by sums. As argued by Moreau et al., however, this normalization is not completely correct. Although a term in $v_j$ can be selected multiple times through the closest-function, it will still be considered only once in the normalization factor. Thus, the similarity computed by $\varphi_s$ can exceed 1. Moreover, a term in the second attribute-value can be matched by multiple terms of the first attribute-value, but not vice versa. Hence, the soft-tf·idf metric is not symmetric, although the Jaro-Winkler metric is (i.e., $\varphi_{\text{jw}}(t_i, t_j) = \varphi_{\text{jw}}(t_j, t_i)$). We thus apply

$$\varphi_{\text{soft-tfidf}}(v_i, v_j) = \min\left(\frac{\varphi_s(v_i, v_j) + \varphi_s(v_j, v_i)}{2}, 1\right)$$

$$\begin{aligned}
\varphi_s(v, v') &= \phi_s(\text{"Havrad"}, v, \text{"Harvard"}, v') + \phi_s(\text{"University"}, v, \text{"Harvard"}, v') \\
&= w_s(\text{"Havrad"}, v) \cdot w_s(\text{"Harvard"}, v') \cdot \varphi_{\text{jw}}(\text{"Havrad"}, \text{"Harvard"}) + 0 \\
&= \frac{\text{tf·idf}(\text{"Havrad"}, v)}{\sqrt{\text{tf·idf}(\text{"Havrad"}, v)^2 + \text{tf·idf}(\text{"University"}, v)^2}} \cdot 1 \cdot 0.918 \\
&= \frac{\text{idf}(\text{"Havrad"})}{\sqrt{\text{idf}(\text{"Havrad"})^2 + \text{idf}(\text{"University"})^2}} \cdot 0.918
\end{aligned}$$

**Figure B.2:** Example of a calculation of the soft-tf·idf-similarity. The example attribute values are $v$ = "Havrad University" and $v'$ = "Harvard".

in our experiments, which does not change the score if the original metric is symmetric or below 1.

# Bibliography

E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints. *Information Retrieval*, 12(4):461–486, August 2009.

J. Artiles, J. Gonzalo, and S. Sekine. The SemEval-2007 WePS Evaluation: Establishing a Benchmark for the Web People Search Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, SemEval-2007, pages 64–69, Stroudsburg, PA, USA, 2007.

J. Artiles, J. Gonzalo, and S. Sekine. WePS2 Evaluation Campaign: Overview of the Web People Search Clustering Task. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-2009), held in conjunction with the 18th International World Wide Web Conference (WWW-2009)*, 2009.

J. Artiles, A. Borthwick, J. Gonzalo, S. Sekine, and E. Amigó. WePS-3 Evaluation Campaign: Overview of the Web People Search Clustering and Attribute Extraction Tasks. In M. Braschler, D. Harman, and E. Pianta, editors, *CLEF 2010 LABs and Workshops, Notebook Papers*, 2010.

A. Bagga and B. Baldwin. Entity-based Cross-document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL-1998) - Volume 1*, pages 79–85, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

K. Balog, L. Azzopardi, and M. de Rijke. Personal Name Resolution of Web People Search. In *Workshop NLP Challenges in the Information Explosion Era (NLPIX-2008), held in conjunction with the International World Wide Web Conference (WWW-2008)*, 2008.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a Mahalanobis Metric from Equivalence Constraints. *J. Mach. Learn. Res.*, 6:937–965, December 2005.

S. Basu, A. Banerjee, and R. J. Mooney. Active Semi-Supervision for Pairwise Constrained Clustering. In M. W. Berry, U. Dayal, C. Kamath, and D. B. Skillicorn, editors, *Proceedings of the 4th SIAM International Conference on Data Mining (SDM-2004)*, pages 333–344, Lake Buena Vista, Florida, April 2004. SIAM.

I. Bhattacharya and L. Getoor. Collective Entity Resolution in Relational Data. *ACM Transactions on Knowledge Discovery from Data*, 1(1):1–36, March 2007.

M. Bilenko, S. Basu, and R. J. Mooney. Integrating Constraints and Metric Learning in Semi-Supervised Clustering. In *Machine Learning, Proceedings of the Twenty-First Internationl Conference (ICML 2004)*. ACM, 2004.

Y. Chen, S. Y. M. Lee, and C. Huang. PolyUHK: A Robust Information Extraction System for Web Personal Names. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-2009), held in conjunction with the 18th International World Wide Web Conference (WWW-2009)*, 2009.

W. Cohen, P. Ravikumar, and S. Fienberg. A Comparison of String Distance Metrics for Name-matching Tasks. In C. Knoblock and S. Kambhampati, editors, *Proceedings of IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-2003)*, pages 73–78, Acapulco, Mexico, August 2003.

I. Davidson and S. S. Ravi. Clustering with Constraints: Feasibility Issues and the k-Means Algorithm. In H. Kargupta, J. Srivastava, C. Kamath, and A. Goodman, editors, *Proceedings of the 5th SIAM International Conference on Data Mining (SMD-2005)*. Society for Industrial Mathematics, 2005.

I. Davidson, K. Wagstaff, and S. Basu. Measuring Constraint-Set Utility for Partitional Clustering Algorithms. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2006)*, volume 4213 of *Lecture Notes in Computer Science*, pages 115–126, Berlin, Germany, 2006. Springer. ISBN 3-540-45374-1.

E. Elmacioglu, Y. F. Tan, S. Yan, M. Y. Kan, and D. Lee. PSNUS: Web People Name Disambiguation by Simple Clustering with Rich Features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 268–271, Prague, Czech Republic, 2007. Association for Computational Linguistics.

A. D. Gordon. Classification in the Presence of Constraints. *Biometrics*, 29: 821–827, 1973.

D. F. Gordon and M. Desjardins. Evaluation and Selection of Biases in Machine Learning. *Machine Learning*, 20:5–22, July 1995.

M. Ikeda, S. Ono, I. Sato, M. Yoshida, and H. Nakagawa. Person Name Disambiguation on the Web by TwoStage Clustering. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-2009), held in conjunction with the 18th International World Wide Web Conference (WWW-2009)*, 2009.

L. Jiang, W. Shen, J. Wang, and N. An. GRAPE: A System for Disambiguating and Tagging People Names in Web Search. In *Proceedings of the 19th International World Wide Web Conference (WWW-2010)*, WWW '10, pages 1257–1260, New York, NY, USA, 2010. ACM.

D. Klein, S. D. Kamvar, and C. D. Manning. From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering. In C. Sammut and A. G. Hoffmann, editors, *Machine Learning, Proceedings of the Nineteenth Internationl Conference (ICML 2002)*, pages 307–314. Morgan Kaufmann, 2002.

J. Kleinberg. An Impossibility Theorem for Clustering. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, pages 446–453. MIT Press, 2002.

G. N. Lance and W. T. Williams. A General Theory of Classificatory Sorting Strategies: I. Hierarchical Systems. *The Computer Journal*, 9:373–380, 1967.

C. Long and L. Shi. Web Person Name Disambiguation by Relevance Weighting of Extended Feature Sets. In M. Braschler, D. Harman, and E. Pianta, editors, *CLEF 2010 LABs and Workshops, Notebook Papers*, 2010.

G. Mann. *Multi-Document Statistical Fact Extraction and Fusion*. PhD thesis, Johns Hopkins University, Baltimore, Maryland, 2006.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

T. M. Mitchell. The Need for Biases in Learning Generalizations. Technical report, Rutgers Computer Science Department, Piscataway, New Jersey, 1980.

E. Moreau, F. Yvon, and O. Cappé. Robust Similarity Measures for Named Entities Matching. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, pages 593–600, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

I. Nagy and R. Farkas. Person Attribute Extraction from the Textual Parts of Web Pages. In M. Braschler, D. Harman, and E. Pianta, editors, *CLEF 2010 LABs and Workshops, Notebook Papers*, 2010.

R. Nuray-Turan, Z. Chen, D. Kalashnikov, and S. Mehrotra. Exploiting Web Querying for Web People Search in WePS2. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-2009), held in conjunction with the 18th International World Wide Web Conference (WWW-2009)*, 2009.

M. F. Porter. *An Algorithm for Suffix Stripping.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

C. J. Van Rijsbergen. Foundation of Evaluation. *Journal of Documentation*, 30(4):365–373, 1974.

S. Salvador and P. Chan. Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '04, pages 576–584, Washington, DC, USA, 2004. IEEE Computer Society.

S. Sekine and J. Artiles. WePS2 Attribute Extraction Task. In *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS-2009), held in conjunction with the 18th International World Wide Web Conference (WWW-2009)*, 2009.

A. Spink, B. Jansen, and J. Pedersen. Searching for People on Web Search Engines. *Journal of Documentation*, 60:266–278, 2004.

L. Talavera and J. Béjar. Integrating Declarative Knowledge in Hierarchical Clustering Tasks. In *Proceedings of the International Symposium on Intelligent Data Analysis*, pages 211–222. Springer-Verlag, 1999.

K. Wagstaff. *Intelligent Clustering with Instance-level Constraints.* PhD thesis, Cornell University, Ithaca, NY, USA, 2002.

K. Wagstaff and C. Cardie. Clustering with Instance-level Constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110, 2000.

X. Wan, J. Gao, M. Li, and B. Ding. Person Pesolution in Person Search Results: WebHawk. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 163–170, New York, NY, USA, 2005. ACM.

W. E. Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research*, pages 354–359, 1990.

W. E. Winkler. Overview of Record Linkage and Current Research Directions. Technical report, Bureau of the census, Washington, DC, 2006.