Integration of Text Mining Taxonomies

Katja Pfeifer(✉) and Eric Peukert

SAP AG, Chemnitzer Str. 48, 01187 Dresden, Germany
{katja.pfeifer01,eric.peukert}@sap.com

Abstract. Text mining services can be used to extract and categorize entities from textual information on the web. Merging results from multiple services could improve extraction quality. This requires to have an integrated extraction taxonomy and corresponding mappings between individual taxonomies that are used for categorizing extracted information. However, current ontology matching approaches cannot be applied since the available meta data within most taxonomies is weak.

In this article we propose a novel taxonomy alignment process that allows us to automatically identify equal, hierarchical and associative mappings and integrate those mappings in a global taxonomy. We broadly evaluate our matching approach on real world service taxonomies and compare to state-of-the-art approaches.

Keywords: Instance-based matching · Text mining · Taxonomy alignment

1 Introduction

Analysts estimate that up to 80 % of all business relevant information within companies and on the web is stored as unstructured textual documents [1]. Being able to exploit such information for example for market analysis, trending or web monitoring is a competitive advantage for companies. To support the extraction of information from unstructured text, a multitude of text mining techniques were proposed in literature (see [2]) and some were publicly made available as Web Services (e.g., [3,4]). These services are able to classify text documents, recognize entities and relationships or identify sentiments. Individual services often have specific strengths and weaknesses. By combining them the overall extraction quality and amount of supported features can be increased [5].

Unfortunately, merging the results from multiple extraction services is problematic since individual services rely on different taxonomies or sets of categories to classify or annotate the extracted information (e.g., entities, relations, text categories). To illustrate the problem we show the results of extracting entities from a news text in Fig. 1.

Entities have been annotated by several text mining services (OpenCalais [3], Evri [6], AlchemyAPI [4], FISE [7]) that rely on different taxonomies to annotate found entities. For instance the text sequence *Airbus* is annotated with three

A. Fred et al. (Eds.): IC3K 2013, CCIS 454, pp. 39–55, 2015.
DOI: 10.1007/978-3-662-46549-3_3

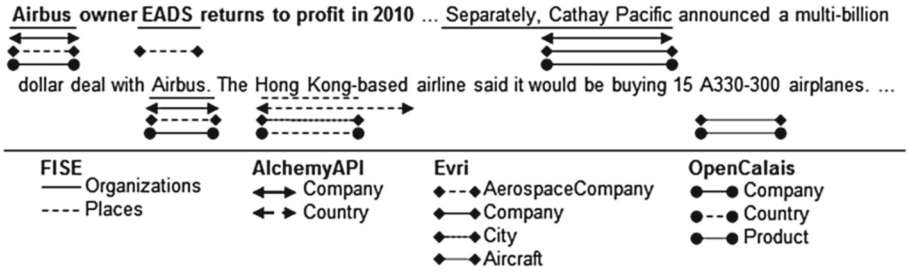


Fig. 1. Analysis of a business news by several named entity recognition services (retrieved on March 9, 2011).

different entity types: *Organization* (by FISE), *Company* (by AlchemyAPI and OpenCalais) and *AerospaceCompany* (by Evri).

To be able to combine and merge extraction results from multiple services a mapping between different taxonomy types and a merged taxonomy is required. Finding such mappings manually is not feasible as the taxonomies can be very large and evolve over time (e.g., AlchemyAPI uses a taxonomy with more than 400 entity types). Unfortunately applying existing (semi-)automatic ontology and schema matching techniques [8,9] does not provide the requested quality since the available meta data within existing service taxonomies is weak (i.e., no descriptions are available, the taxonomies have a flat structure). Moreover, existing matching approaches are not able to identify relations between the taxonomy types (i.e., if two types are equal or if one type is a subtype of the other).

To overcome those limitations, we introduce a novel taxonomy alignment process that enables the merging of taxonomies for text mining services. The following contributions are made within this article:

- We introduce a novel taxonomy alignment approach that is based on generated instances of input taxonomies.
- In particular, a novel metric for instance-based matchers is proposed that is able to identify equal, hierarchical and associative mappings.
- Based on the automatically computed mappings a cluster-based taxonomy merging process is described.
- The taxonomy alignment process is compared to state-of-the-art instance-based alignment methods. For evaluation, reference mappings between a number of real-world text mining services and their taxonomies were created through an online survey with numerous participants.

The remainder of the article is structured as follows: In Sect. 2 we formally describe the problem and introduce the notation being used within this article. Section 3 introduces our taxonomy alignment process and presents the metric for instance-based matching as well as the combined matching strategy used within our process. The experimental setup and the results of our evaluation can be found in Sects. 4 and 5. We introduce a taxonomy merging approach that makes use of the introduced taxonomy alignment process in Sect. 6 before we review related work in Sect. 7. Section 8 closes with conclusions and an outlook to future work.

2 Problem Description

Combining the results of multiple text mining services is promising as it can increase the quality and functionality of text mining. This requires us to have a mapping between the underlying taxonomies of the individual extraction services. However, finding such a mapping is challenging. A review of existing text mining services and their taxonomies revealed that the taxonomies differ strongly in granularity, naming and their modeling style. Many taxonomies are only weakly structured and most taxonomy types are lacking any textual description. Therefore manually defining a mapping between text-mining taxonomies is a complex, challenging and time consuming task.

Within this article we want to apply ontology- and schema matching techniques [8,9] to automatically compute mappings between text mining taxonomies. Matching systems take a source and a target ontology as input and compute mappings (alignments) as output. They employ a set of so called matchers to compute similarities between elements of the source and target and assign a similarity value between 0 and 1 to each identified correspondence. Some matchers primarily rely on schema-level information whereas others also include instance information to compute element similarities. Typically, the results from multiple of such matchers are combined by an aggregation operation to increase matching quality. In a final step a selection operation filters the most probable correspondence to form the final alignment result.

Unfortunately existing matching approaches solve the challenges of matching text mining taxonomies only partly. Schema-based matchers can only be applied to identify mappings between equal concepts (e.g., by using a name-matcher) as the scarcity of broader meta data disables the use of more enhanced matchers (e.g., retrieving hierarchical mappings through the comparison of the taxonomy structure). Instance-based approaches are mainly limited to equal mappings. The few instance-based approaches that support hierarchical mappings still suffer from limited accuracy as we show in our evaluation (see Sect. 7 for a broader review of related work).

To overcome the aforementioned limitations, we proposed an instance enrichment algorithm in [10] that populates the taxonomy types with meaningful instances. This allows us to apply instance-based matchers and similarity metrics like Jaccard and Dice [11,12] to identify mapping candidates. Since those metrics can only be used to identify equality mappings we introduce a novel metric that allows to identify hierarchical and associative mappings like broader-than, narrower-than or is-related to. We integrate the instance enrichment and instance matching together with some optimizations in a novel taxonomy alignment process that we describe below.

To sharpen the description of our contributions, we formalize the problem. The overall goal of the taxonomy alignment process is to integrate the taxonomies $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n$ of the text mining services $\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_n$ into one global taxonomy $\mathcal{G}$. We make the assumption that each service $\mathcal{S}_i$ uses its own taxonomy $\mathcal{T}_i$ to classify the text mining results. In order to align two taxonomies $\mathcal{T}_s$ and $\mathcal{T}_t$ mappings between the types of the taxonomies need to be identified.

A mapping M is a triple (T_{sj}, T_{tk}, R) in which $R \in \{\equiv, <, >, \sim\}$ indicates a relation between a type $T_{sj} \in \mathcal{T}_s$ and a type $T_{tk} \in \mathcal{T}_t$. $(T_{sj}, T_{tk}, \equiv)$ means that the taxonomy types T_{sj} and T_{tk} are equivalent, $(T_{sj}, T_{tk}, <)$ indicates that T_{sj} is a subtype of T_{tk} (i.e., T_{sj} is narrower than T_{tk}), $(T_{sj}, T_{tk}, >)$ is the inverse subsumption relation (i.e., T_{sj} is broader than T_{tk}). $(T_{sj}, T_{tk}, \sim)$ represents an associative relation (e.g., *car* and *truck* are associated). The set of instances annotated by a type T_{ij} is specified by $I(T_{ij})$, its cardinality by $|I(T_{ij})|$. When matching two dissimilar taxonomies we speak of inter-matching whereas matching the types of a taxonomy with itself ($\mathcal{T}_s = \mathcal{T}_t$) is called intra-matching. Since equal mappings are not relevant in the intra-matching case the set of relevant relations is $R \in \{<, >, \sim\}$.

3 Taxonomy Alignment Process

Initially, the overall taxonomy alignment process is described. Section 3.2 introduces our new metric that is applied within the alignment process. The matching process applying instance- and schema-based matching is detailed in Sect. 3.3.

3.1 Overall Alignment Process

The general taxonomy alignment process is depicted in Fig. 2. The overall idea is to retrieve mappings for the taxonomy types by a matching process and subsequently integrate those mappings to form a global taxonomy $\mathcal{G}$ (see Sect. 6 for the integration). This taxonomy $\mathcal{G}$ reflects all types of the individual taxonomies $\mathcal{T}_i$ and the relations between the particular types (expressed in the mappings). Additionally, the mappings can optionally be cleaned (e.g., by detecting cycles within the graph) and complemented by new mappings (e.g., by exploiting the given hierarchical structure) in mapping rewrite steps as done by existing ontology matching tools like ASMOV [13]. In order to integrate n taxonomies $\binom{n}{2}$ inter-matching processes and n intra-matching processes are applied within our taxonomy alignment process. Each of these inter-matching processes takes two taxonomies as input and identifies equivalence, hierarchical and associative mappings between the types of these taxonomies. The intra-matching processes discover hierarchical and associative mappings within one taxonomy in order to validate and correct/enhance the existing taxonomy structures.

The inter-matching process is implemented by a combined matcher consisting of a schema-based and an instance-based matcher. The schema-based matcher exploits the names of the taxonomy types (e.g., $T_1.a$ and $T_2.i$ in Fig. 2) and is able to identify candidates for equivalence mappings. If sufficient meta data is available for the taxonomies, the schema-based matcher can be extended with matchers that additionally take into account the descriptions or the structures of the input taxonomies. The instance-based matcher exploits the instances of the taxonomy types to identify mapping candidates. The instances of the taxonomy types are retrieved by a new iterative instance enrichment algorithm that was presented in [10]. Furthermore the instance-based matcher applies a novel

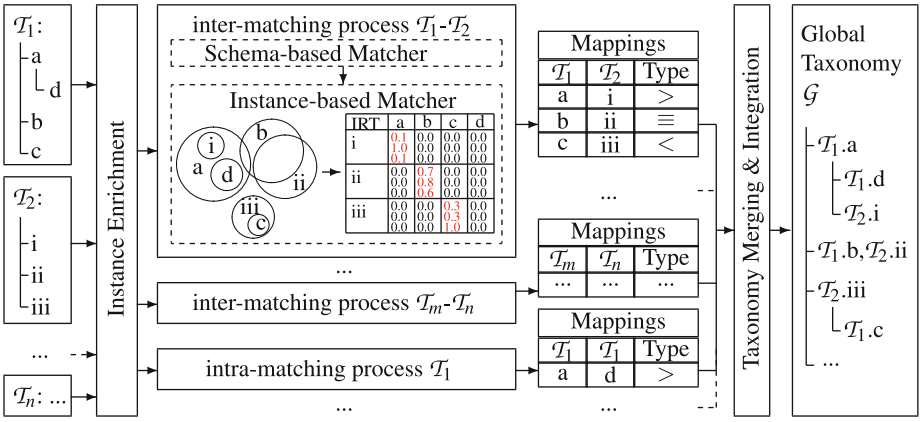


Fig. 2. Taxonomy alignment process.

similarity metric – the intersection ratio triple (IRT) – that allows to identify equivalence, hierarchical as wells as associative relations between the taxonomy types. We will present the metric in Sect. 3.2 and give details on the inter- and intra-matching process in Sect. 3.3.

The intra-matching process uses a slightly adjusted version of the instance-based matcher. A combination with a schema-based matcher is not necessary as equivalence mappings are irrelevant here. The results of the intra-matching process can be used to bring structure into flat taxonomies and check and correct given taxonomy structures.

3.2 IRT Metric

In this section, we present our novel similarity metric for instance-based matchers that is able to indicate equivalence, hierarchical and associative relations between the elements of two taxonomies $\mathcal{T}_s$ and $\mathcal{T}_t$. Additionally it allows to identify hierarchical and associative relations within one taxonomy, when used with slightly changed parameters.

It is a common technique within instances-based matchers to rate the similarity of two taxonomy elements $T_{sj} \in \mathcal{T}_s$ and $T_{tk} \in \mathcal{T}_t$ by analyzing instance overlaps and to represent them by a similarity metric. We propose a novel metric that consists of three single values to represent equivalence, hierarchical and associative relations. The metric adopts the corrected Jaccard coefficient presented by [11]:

$$JCcorr(T_{sj}, T_{tk}) = \frac{\sqrt{|I(T_{sj}) \cap I(T_{tk})| \times (|I(T_{sj}) \cap I(T_{tk})| - c)}}{|I(T_{sj}) \cup I(T_{tk})|}$$

In contrast to the original Jaccard coefficient, that is the ratio of the instance intersection size and the size of the union of the instances, the corrected Jaccard

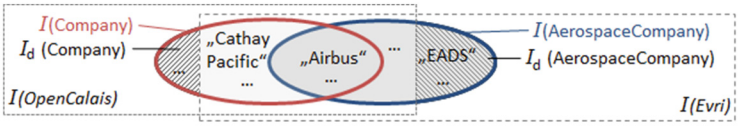


Fig. 3. Example for quality restrictions.

coefficient considers the frequency of co-occurring instances with its correction factor c. For details how to configure c please refer to [11].

We rely on this basic metric as it allows us to deal with possible data sparseness of the instances determined with our instance enrichment process. Additionally, the instances retrieved from text mining services have some quality restrictions that need to be handled. Text mining faces the problem of potentially being inaccurate. Thus, the instances can include false positives (i.e., instances having been extracted wrongly) and for some services miss false negatives (e.g., instances that should be extracted, but having eventually only been extracted by some services).

In order to handle these quality restrictions, we propose an extension of the corrected Jaccard metric as follows: We introduce a weakening factor w that reduces a negative effect of instances only found by one of the services. The factor is trying to correct the influence of the false positives and negatives of the extraction process. Therefore the set of distinct instances $I_d(T_{sj})$ and $I_d(T_{tk})$ that were only extracted by one of the services (independent from the entity type assigned to them) are integrated in the corrected Jaccard factor weakened by w:

$$JCcorr^{+}\left(T_{sj}, T_{tk}\right) = \frac{\sqrt{\left|I(T_{sj}) \cap I(T_{tk})\right| \times \left(\left|I(T_{sj}) \cap I(T_{tk})\right| - c\right)}}{\left|I(T_{sj}) \cup I(T_{tk})\right| - w\left|I_d(T_{sj})\right| - w\left|I_d(T_{tk})\right|}$$

$$\text{with } I_d(T_{sj}) \subseteq I(T_{sj}) \backslash \bigcup_{A \in \mathcal{T}_t} I(A), I_d(T_{tk}) \subseteq I(T_{tk}) \backslash \bigcup_{B \in \mathcal{T}_s} I(B) \text{ and } 0 \leq w \leq 1$$

Figure 3 exemplarily depicts the interrelationships between the quality restrictions (e.g., “EADS” as false negative annotation for OpenCalais) and the distinct instances (data was retrieved from Fig. 1).

The similarity value retrieved by the $JCcorr^{+}$ coefficient enables decisions on the equality of two taxonomy types. If the value is close to 1 it is likely that the type T_{sj} is equal to T_{tk}, if the value is 0, the two taxonomy types seem to be unequal. However, the similarity value does not provide an insight into the relatedness of the two types, when the value is neither close to 1 nor 0. Let us consider the type *Company* and the type *AerospaceCompany*. The extended corrected Jaccard value would be very small – only those company instances of the *Company* type that are aerospace companies might be in the intersection, whereas the union set is mainly determined by the instance size of the type *Company*. In order to detect subtype and associative relations we introduce two

more measures $JCcorr^{+}_{T_{sj}}$ and $JCcorr^{+}_{T_{tk}}$ rating the intersection size per type:

$$JCcorr^{+}_{T_{sj}}(T_{sj}, T_{tk}) = \frac{\sqrt{|I(T_{sj}) \cap I(T_{tk})| \times (|I(T_{sj}) \cap I(T_{tk})| - c)}}{|I(T_{sj})| - w\,|I_d(T_{sj})|}$$

$$JCcorr^{+}_{T_{tk}}(T_{sj}, T_{tk}) = \frac{\sqrt{|I(T_{sj}) \cap I(T_{tk})| \times (|I(T_{sj}) \cap I(T_{tk})| - c)}}{|I(T_{tk})| - w\,|I_d(T_{tk})|}$$

These coefficients are the ratio of the intersection size of the instance sets of the two elements T_{sj} and T_{tk} and the size of one of the instance sets (the instance set $I(T_{sj})$ and $I(T_{tk})$ respectively). All three intersection values together ($JCcorr^{+}, JCcorr^{+}_{T_{sj}}, JCcorr^{+}_{T_{tk}}$) form the intersection ratio triple (IRT). We can monitor the following states for the values of the IRT metric:

- If all three values are very high, it is very likely that the elements for which the measures were calculated are equal, i.e., the mapping $(T_{sj}, T_{tk}, \equiv)$ can be derived.
- If $JCcorr^{+}_{T_{sj}}$ is high and the difference $\text{diff}_{T_{tk}}$ of $JCcorr^{+}$ and $JCcorr^{+}_{T_{tk}}$ is close to zero, it is an indication that the element T_{sj} is a subtype of T_{tk}, i.e., the mapping $(T_{sj}, T_{tk}, <)$ can be derived.
- If $JCcorr^{+}_{T_{tk}}$ is high and the difference $\text{diff}_{T_{sj}}$ of $JCcorr^{+}$ and $JCcorr^{+}_{T_{sj}}$ is close to zero, it is an indication that the element T_{tk} is a subtype of T_{sj}, i.e., the mapping $(T_{sj}, T_{tk}, >)$ can be derived.
- If none of the three states above yields, but at least one of the IRT-values is clearly above zero the elements T_{sj} and T_{tk} are associated, i.e., the mapping $(T_{sj}, T_{tk}, \sim)$ can be derived.

The IRT metric can also be applied for intra-matching processes. However, the weighting factor is set to 0, i.e., the corrected Jaccard coefficient (and the modified corrected Jaccard coefficients for the second and the third value of the IRT) is used in fact. In the following we show how our novel metric is used within our combined matcher.

3.3 The Matching Process

As already described we use a complex matching strategy that combines both schema-based and instance-based matcher in a single matching process. The combination strategy is visualized in Fig. 4.

The strategy consists of a number of operators that are commonly used in schema matching such as selection (Sel), aggregation (Agg) and matching (mat). Moreover two additional operators ($Trans$ and $Diff$) are included that are needed for processing the IRT matcher results. The process starts by executing the schema- and our instance-based matcher (mat_{schema} and mat_{inst}). They take as input the two taxonomies $\mathcal{T}_s$ and $\mathcal{T}_t$ and calculate a similarity matrix consisting of $|\mathcal{T}_s| \times |\mathcal{T}_t|$ entries (Sim and Sim_{IRT}). Each entry of the Sim-matrix is a value between 0 and 1 with 0 representing low and 1 representing high similarity between two pairs of elements from the input taxonomies. The similarity values of this matrix are calculated by a simple name-matcher as proposed

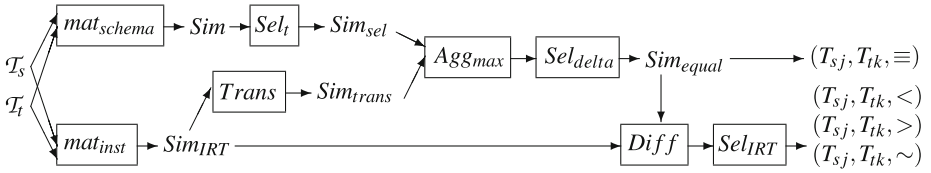


Fig. 4. Combined matching strategy.

in COMA++ [14]. In contrast to that, the entries of the Sim_{IRT}-matrix are composed of the three values computed by our IRT metric (see an exemplary IRT-matrix in Fig. 2).

For equal mappings, we trust in the most likely matching candidates identified by the schema-based matcher. As discussed, the naming of taxonomy types is typically clear and precise and therefore name-matchers tend to have a very high precision. With a selection operation Sel_t the most probable matching candidates are extracted. This operation sets all matrix entries below a given threshold to 0 and all others to 1. We pick a high selection threshold (0.8) to minimize the chance to select wrong mappings.

To simplify the combination of the Sim_{IRT} matrix and the Sim_{sel} matrix, the Sim_{IRT} matrix is transformed by a transformation operation $Trans$. It maps the three IRT values to one value that expresses the probability that the two taxonomy elements are equal. Different transformation operations are possible. A trivial transformation operation $trans_{triv}$ just takes the first IRT value (the extended corrected Jaccard coefficient $JCcorr^+$) or the average of all three values. However, such a trivial transformation may lead to false positive equal mappings since some identified candidates may rather be subtype mappings. As already mentioned in Sect. 3.2 a very low difference value $\text{diff}_{T_{sj}}$ and $\text{diff}_{T_{tk}}$ respectively, may indicate a hierarchical relation. We therefore propose a transformation that lowers the similarity values for such cases:

$$trans = trans_{triv} - corr_{sub}$$

$$corr_{sub} = \begin{cases} 0 & \text{max diff of IRT values} < 0.2 \\ z \cdot e^{-\lambda \cdot \text{diff}_{T_{sj}}} & JCcorr^+_{T_{sj}} < JCcorr^+_{T_{tk}} \\ z \cdot e^{-\lambda \cdot \text{diff}_{T_{tk}}} & JCcorr^+_{T_{sj}} > JCcorr^+_{T_{tk}} \end{cases} \quad \text{with } \lambda > 0 \text{ and } 0 \leq z \leq 1$$

The transformation relies on an exponential function to weight the influence of the difference values ($\text{diff}_{T_{sj}}$ or $\text{diff}_{T_{tk}}$) on the transformation result. In particular when the three IRT values are not very close to each other (i.e., having a maximal difference greater than 0.2) the exponential function is applied. The subtype correction $corr_{sub}$ has the biggest value if the difference is zero and then exponentially decreases to zero. The λ value defines how strong the value decreases. Example: With $\lambda = 20$ and a difference value of 0.05 the value $trans_{triv}$ is decreased by 0.368. For $\lambda = 100$ the decrease is only 0.007. The correction value can be further adapted by a weight z that can be based on the value of $JCcorr^+_{T_{sj}}$ and $JCcorr^+_{T_{tk}}$ respectively.

The selected similarity matrix Sim_{sel} is combined with the transformed similarity matrix Sim_{trans} of the instance-based matcher with a MAX-Aggregation operation Agg_{max}. For each pair of entity pairs the maximum of the two matrix entries (one entry from the Sim_{sel} and one from Sim_{trans} matrix) is taken. The result of the mapping aggregation still contains up to $|\mathcal{T}_s| \times |\mathcal{T}_t|$ correspondences. From these correspondences the most probable ones need to be selected. A number of selection techniques have been proposed in literature (see [14]). We apply the MaxDelta selection from [14] in Sel_{delta} since it has shown to be an effective selection strategy. MaxDelta takes the maximal correspondence within a row (or column) of a similarity matrix. Additionally, it includes correspondences from the row (or column) that are within a delta-environment of the maximal correspondence. The size of the delta environment depends on the value of the maximal element for each row (or column). Both sets of maximal correspondences for each row and correspondences for each column are intersected to get the final selection result Sim_{equal}. Finally, equality mappings are created from the selected matrix Sim_{equal} for each matrix entry above a given threshold.

Subtype and associative mappings are directly derived from the Sim_{IRT} matrix. However, all equality mapping candidates are eliminated from the matrix ($Diff$) before a fine granular selection operation Sel_{IRT} is applied. Sel_{IRT} derives subtype mappings if $JCcorr^{+}_{T_{sj}}$ (or $JCcorr^{+}_{T_{tk}}$) is above a given threshold and if $\text{diff}_{T_{tk}}$ (or $\text{diff}_{T_{sj}}$) is smaller than a distance threshold. All remaining matrix entries that are not selected as subtype mappings but indicate a certain overlap of the instances are categorized as associative mappings if one of the three IRT values is significantly above zero.

The presented strategy can be adaptively fine-tuned by analyzing the results of the schema-based matcher. Differing strength and performance of the extraction services for which taxonomies are matched can be identified. For instance, if the text mining service $\mathcal{S}_s$ is consistently stronger than the service $\mathcal{S}_t$, we can observe the following: The instance set $I(T_{tk})$ is included in the instance set $I(T_{sj})$ even if the two taxonomy types T_{sj} and T_{tk} are identical (i.e., the schema-based matcher indicates an equivalence relation). For those cases a transformation which corrects subtypes is not recommended. Additionally the selection thresholds can be adapted by observing the instance-matching values for which equivalence relations hold.

4 Experimental Setup

Before we present the results of our experiments in matching entity taxonomies of text mining services in Sect. 5, we give an overview of the experimental setup. The goal of the experiments was to evaluate if our automatic matching approach is applicable for matching taxonomies of text mining services and if our novel metric performs better than traditional approaches. All datasets and manually created gold standards are available upon request.

4.1 Dataset

We evaluated our approach on three entity taxonomies of public and well known text mining services, that are OpenCalais [3], AlchemyAPI [4] and Evri [6]. We only considered the taxonomies that are provided for English text. The entity taxonomy of OpenCalais is documented on the service website and consists of 39 main entity types that are partially further specified with predefined attributes (e.g., the entity Person has the attributes PersonType, CommonName, Nationality). In total it contains 58 entity types. AlchemyAPI documented its entity types classified in a two-level hierarchy on the service website. We observed that not all types AlchemyAPI extracts are listed on the service website. That is why we extended the taxonomy with types having been extracted during the instance enrichment process. All together the taxonomy then consists of 436 types. Evri does not provide an overview of the entity types the service can extract. However, it was possible to extract information via service calls. The Evri taxonomy constructed from the service calls is made up of 583 types.

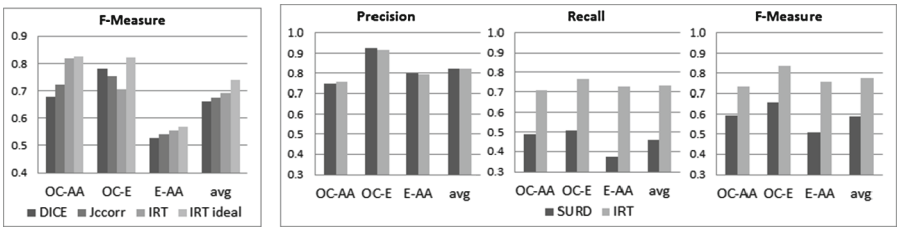
4.2 Gold Standard

So far no mappings between the taxonomies of text mining services existed. In order to evaluate the quality of the mappings retrieved with our approach, we manually produced a gold standard in [10] through an online evaluation.

We use three values to rate the quality of the retrieved mappings compared to the gold standard: precision, recall and F-measure. Precision is the ratio of accurately identified mappings (i.e., the ratio of the retrieved mappings being in the gold standard and the retrieved mappings). Recall marks the ratio of mappings within the gold standard that were identified by the matcher. The F-measure is the harmonic mean of precision and recall and is a common metric to rate the performance of matching techniques. We consider a matcher to be as good as the F-measure is.

4.3 Matcher Configurations

We experimented with different configurations of our instance-based matcher and determined the best setting - a Jaccard correction factor $c = 0.6$ and a weight w to 0.95 (i.e., integrated the instances only retrieved by one of the services to five percent into the calculations). We achieved good results with a transformation operation using the average of the three IRT values slightly corrected by the exponential function as given in Sect. 3.3. We scaled this correction down or rather ignored it, when observing strongly differing service strength (that was the case, when matching the taxonomy of the OpenCalais service with the taxonomies of the weaker services AlchemyAPI and Evri). The selection threshold for retrieving equality mappings was set to 0.2 when used stand alone and to 0.5 when used in the combined matcher. For the subtype selection operation we used a threshold of 0.65 and a distance threshold of 0.05 within inter-matching processes and a threshold of 0.9 and 0.001 within intra-matching processes.



(a) Equal (b) Subtype

Fig. 5. Comparison of similarity metrics.

We compared our instance-based matching approach and the IRT metric to common metrics of instance-based matching systems: for equality mappings we compared against the Dice and the corrected Jaccard metric, for hierarchical mappings against the SURD metric. The selection thresholds of Dice and corrected Jaccard were set to those values for which the highest average F-measure could be retrieved (Dice: 0.1, corrected Jaccard with correction factor 0.8: 0.05). For SURD we used the threshold proposed in [15] – ratios below 0.5 are low values, ratios above 0.5 are high values. Independent from the used metric the instance intersections were determined by comparing the strings of the instances and only accepting exact matches for the intersection. Moreover, the Sel_{delta} selection techniques described in Sect. 3.3 was applied in all cases.

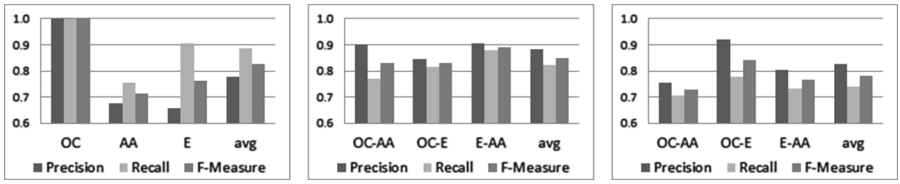
5 Experimental Results

In the following we present our experimental results proving that our approach is applicable for matching taxonomies of text mining services. We start comparing the IRT metric to state-of-the-art metrics for instance-based matching in Sect. 5.1. Afterwards we rate the performance of the overall intra- and inter-matching processes in Sect. 5.2.

5.1 Comparison of Similarity Metrics

We compared the IRT metric to Dice, corrected Jaccard and SURD and analyzed the performance regarding the identification of equal and subtype mappings (see Sect. 4.3 for the matcher configurations). The results of the comparison are depicted in Fig. 5, in which *OC-AA* indicates the matching process between the OpenCalais and the AlchemyAPI taxonomy, *OC-E* between OpenCalais and Evri, *E-AA* between Evri and AlchemyAPI and *avg* the average between the three values.

Figure 5(a) shows the F-Measure for retrieving equality mappings. We were able to slightly increase the average F-measure compared to the classical metrics Dice and corrected Jaccard. When individually setting the threshold (e.g., by using the schema-based matcher as indicator) the F-measure as well as precision



(a) intra matching (b) inter matching equal (c) inter matching subtype

Fig. 6. Performance of our matching approach.

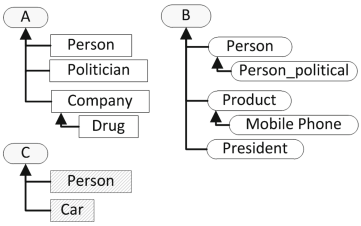
and recall can be again increased (IRT ideal). Independent from the specific metric used the performance for the matching process between Evri and AlchemyAPI is worse than the other two matching processes. Reasons for this are on the one hand relatively few instances used for the matching and on the other hand the big performance difference of the two services. We detected that in average equal types only have 30 % in common and it is therefore very hard to detect all mappings correctly.

Figure 5(b) presents the results for the identification of subtype mappings. One can see, that the IRT metric can significantly raise the recall (nearly 30 %) by keeping the same good precision like the SURD metric. Thereby the F-measure can be increased by nearly 20 % which proves that our IRT metric is suited much better for the matching of text mining taxonomies.

5.2 Overall Matching Process

We applied the instance enrichment algorithm, the IRT metric and the combined matching strategy for the intra- and the inter-matching processes for the three services and their taxonomies. The performance results are given in Fig. 6. We compared the mapping results of the intra-matcher to the relations given within the taxonomy structure (Fig. 6(a)). Our approach covered exactly the relations given within the OpenCalais taxonomy. On the contrary, the mappings retrieved by our matching approach and the relations of the AlchemyAPI and Evri taxonomy differed. However, this discrepancy is not a result of the inability of our approach, but rather an indication that the taxonomies are not structured accurately. *AircraftDesigner* is for example listed as a *Person* subtype in the taxonomy used by AlchemyAPI. In practice aircraft designing companies instead of persons are annotated with this type. On the other hand, the flat structure of the taxonomies ignores relations within the subtypes of an entity. *USPresident* and *Politician* are both subtypes of *Person* (which is given in the taxonomy) and the former is in addition a subtype of the latter (this information was retrieved by our approach, but is not represented in the taxonomy). The results show that overreliance on the given taxonomy structures is not reasonable. Instead our approach should be used to validate and correct the taxonomy structure.

The results for the inter-matching processes clearly show that a combination of schema- and instance-based matcher improves quality. The F-measure has



Taxonomy Mappings

	Source	Target	Relation	Confidence
Equal	A.Person	B.Person	=	0.8
	A.Politician	B.Person_political	=	0.8
	A.Product	B.Product	=	0.8
Subtype	A.Person	C.Person	>	0.6
	A.Person	B.President	>	0.5
	A.Person	B.Person_political	>	0.5
	A.Politician	B.President	>	0.5
	A.Drug	B.Product	<	0.7
	A.Company	C.Car	>	0.4
	A.Product	C.Car	>	0.7
Intra-Subtype	A.Politician	A.Person	<	0.6
	B.President	B.Person_political	<	0.5
	B.Product	B.Mobile Phone	>	0.6
Related	A.Company	B.Product	~	0.3

Fig. 7. Taxonomy examples and mapping.

been raised by more than 15 % compared to the instance-based matcher only approach (see Fig. 5). An average F-measure of 85 % for equal (Fig. 6(b)) and 77 % for subtype (Fig. 6(c)) shows that an automatic matching of text mining taxonomies is possible. We observed that in average 63 % of the wrong subtype mappings and 16 % of the missed subtype mappings can be traced back to instance scarcity (i.e., have five or less instances in the intersection). One quarter of the missed equal mappings result from instance scarcity too. Increasing the amount of instances (e.g., by allowing more iterations in the instance enrichment process) and adapting the parameters for each matching process separately (e.g., by using the name-matcher as an indication for the thresholds) quality can be increased.

6 Cluster-Based Taxonomy Merging

To be able to make use of the identified taxonomy mappings the individual taxonomies need to be merged to build a global taxonomy. That global taxonomy could serve as an interface to the combined extraction service. Given a number of taxonomies $T_1, T_2, \ldots, T_n$, a mapping between each possible pair of input taxonomy needs to be computed. The example in Fig. 7 consists of three simple taxonomies that need to be merged and the computed set of mappings. Each mapping entry in the table consists of source, target, relation and confidence. The matching process from above is able to retrieve the mappings and assign confidences to each computed mapping entry. Due to the automatic instance enrichment and matching not all possible mappings are in fact identified and some might be incorrect or questionable like *A.Company* > *C.Car*.

In the example three equal mappings have been identified, that is between the *A.Person* category from Taxonomy A and *B.Person* category from Taxonomy B, between *A.Politician* and *B.Person_Political* and between *A.Product* and *B.Product*. Moreover, a number of subtype mappings have been found. In particular with subtype mappings it often happens that a category is a subtype of multiple other categories which is natural. The US-President is both a Person and a Politician. The equal and subtype mappings are complemented by related mappings. Only one related mapping is listed in the example. Finally also intra

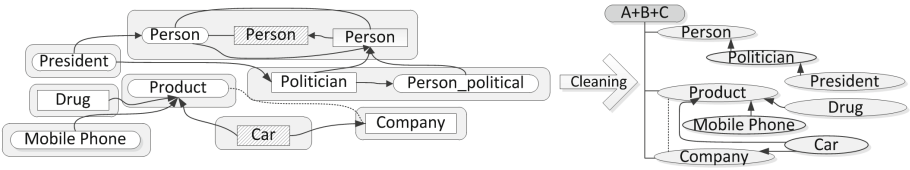


Fig. 8. Clustering and Merging.

mappings were computed that later help to structure the final merged taxonomy and correct the given taxonomy structures. In the following a process is described that builds an integrated global taxonomy that can be exposed for the merged service and that can be used for annotating results.

1. From the found mappings a graph is build. For each type of mapping a different edge is used in the example Fig. 8 (a undirected edge for equal, a directed edge for subtype and a dotted undirected one for associated). Obviously there is still some ambiguity that needs to be resolved and the different categories need to be merged.
2. We identify strongly connected components in the graph of equal matches. The output are clusters of strongly connected elements (illustrated by the grey ovals in the left side of Fig. 8). These clusters build the new categories. Cluster names are assigned from the category that collected most instances in the instance enrichment example.
3. If a cluster also consists of subtype matches then such matches are changed to equal matches.
4. The edges from categories within a cluster pointing to categories outside of a cluster are now changed to edges between the clusters instead of the individual categories. Multiple similar edges between clusters are replaced by one. However the single edge remembers how many and which representatives formally existed.
5. In the next step, the transitivity relation is exploited to remove unnecessary subtype matches between clusters. That means, if a cluster A is subtype B and B is subtype C then all subtype edges between A and C can be removed.
6. The remaining edges between the clusters are further cleaned. If there is a subtype and an equal edge between clusters - then the subtype edge is taken and the equal edge is removed. If there are equal edges between clusters the edge is replaced by related edges. If subtype edges point in both directions then the edge with smaller edge count is removed.

There can be further rules to correct the graph such as removing possible cycles of subtype edges. The final merged taxonomy A+B+C is shown in Fig. 8. Note, that this taxonomy is not a tree, it is rather a graph of categories. The mapping result from above can also be filtered with higher thresholds to remove some mappings with low confidence. The final merged taxonomy structure then has higher credibility. However, the number of categories could increase and the final

taxonomy might be less structured. For each service that should be merged the merged taxonomy needs to be recomputed. This ensures that the order of addition has no influence on the result. Note that for each cluster in the taxonomy, the original cluster is retained as a mapping between the individual taxonomies and the global taxonomy. First evaluations of the cluster-based merging process are promising and a first use case to illustrate the value of computing taxonomy alignments between extraction services was implemented with a web news analysis application [10].

7 Related Work

A number of matching systems have been developed that are able to semi-automatically match meta data structures like taxonomies, ontologies or XSD schemata (see [9, 16]). Most of these systems rely on schema-based matching techniques, that consider names, structure or descriptions of elements for matching. For some test-cases they are able to identify equal mappings as we show in our evaluation. However, schema-based techniques are not suited to generate subtype or associative mappings when dealing with flat taxonomies.

A number of existing matching systems like QuickMig [17], COMA++ [14], RiMOM [18] or Falcon [19] rely on instance-based matching techniques to find further correspondences when schema-based matchers are not sufficient. Some of them look for equality of single instances [17–19], others employ metrics that rely on the overlap of instance sets [14]. The latter rely on similarity metrics like Jaccard, corrected Jaccard, Pointwise Mutual Information, Log-Likelihood ratio and Information Gain (see [11]). Massmann and Rahm [12] apply the dice metric to match web directories from Amazon and Ebay. All of these similarity metrics can only be applied to retrieve equal mappings. Moreover, they only perform well when instance sets are quite similar and strongly intersect. They do not consider inaccurate and incomplete instances, like we do with our IRT metric.

The PARIS system [20] employs a probabilistic approach to find alignments between instances, relations and classes of ontologies. The system is mainly able to identify equivalence relations but the authors also introduce an approach to find subclass relations. However, they neither presented how to apply this approach in order to decide for equivalence or subtype relations of classes nor have they evaluated the identification of subclasses. Reference [15] recently proposed a metric of two coefficients to resolve the question how to identify hierarchical relationships between ontologies. This metric is similar to our IRT metric, but does not consider failures within the instances. Moreover, due to relying on only two values and basic heuristics this metric is more inaccurate than the IRT metric presented in this article. By relying on three coefficients we can further refine relationships and besides identifying equivalence and hierarchical relations also identify associative relations between the types of two taxonomies which can not be done with metrics proposed so far. Moreover, we are the first to apply ontology matching techniques for matching text mining taxonomies.

For merging taxonomies most approaches apply merging based on an algorithm where one structure is integrated into another one like PORSCHE [21] or ATOM [22]. The order of merging with these approaches is crucial and can change the resulting global structure. In contrast to that, cluster-based merging approaches do not rely on the order of merges and always reflect an optimal global schema as was also done in ARTEMIS [23] or by Dragut et al. [24] when integrating web data source schemata. Our approach goes beyond existing work since it includes subtype edges and internal structure in the merge process which creates a global schema of higher quality similar to MOMIS [25].

8 Conclusions and Future Work

In this article we presented a number of contributions that help to automatically match and integrate taxonomies of text mining services and therewith enable the combination of several text mining services. In particular we proposed a general taxonomy alignment process that applies a new instance-based matcher using a novel metric called IRT. This metric allows us to derive equality, hierarchical and associative mappings. Our evaluation results are promising, showing that the instance enrichment and matching approach returns good quality mappings and outperforms traditional metrics. Furthermore, the matching process again indicated that the results of different text mining services are very different, i.e., the instances of semantically identical taxonomy types are only partly overlapping (partly only 5 % of the instances overlap). This emphasizes the results from [5] that the quality and quantity of text mining can be increased through the aggregation of text mining results from different services. The presented taxonomy alignment process will allow us in future to automate the matching of text mining taxonomies and subsequently the automatic merging of text mining results from different services (see [26]).

References

1. Grimes, S.: Unstructured data and the 80 percent rule. Clarabridge Bridgepoints (2008). http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/
2. Hotho, A., Nürnberger, A., Paaß, G.: A brief survey of text mining. LDV Forum **20**(1), 19–62 (2005)
3. OpenCalais: Calais Homepage. March 2013. http://www.opencalais.com/
4. AlchemyAPI: AlchemyAPI Homepage. March 2013. http://www.alchemyapi.com/
5. Seidler, K., Schill, A.: Service-oriented information extraction. In: Proceedings of the Joint EDBT/ICDT Ph.D. Workshop 2011, pp. 25–31 (2011)
6. Evri: Evri Developer Homepage. June 2012. http://www.evri.com/developer/
7. FISE: Furtwangen IKS Semantic Engine project page. March 2013. http://wiki.iks-project.eu/index.php/FISE
8. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, New York (2007)
9. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**, 334–350 (2001)

10. Pfeifer, K., Peukert, E.: Mapping text mining taxonomies. In: KDIR 2013 Proceedings, Scitepress, Portugal (2013)
11. Isaac, A., van der Meij, L., Schlobach, S., Wang, S.: An empirical study of instance-based ontology matching. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 253–266. Springer, Heidelberg (2007)
12. Massmann, S., Rahm, E.: Evaluating instance-based matching of web directories. In: WebDB 2008 Proceedings (2008)
13. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. Web Semant. **7**(3), 235–251 (2009)
14. Do, H.H., Rahm, E.: COMA - a system for flexible combination of schema matching approach. In: VLDB Proceedings (2002)
15. Chua, W.W.K., Kim, J.J.: Discovering cross-ontology subsumption relationships by using ontological annotations on biomedical literature. In: ICBO. CEUR Workshop Proceedings, vol. 897 (2012)
16. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. In: Spaccapietra, S. (ed.) Journal on Data Semantics IV. LNCS, vol. 3730, pp. 146–171. Springer, Heidelberg (2005)
17. Drumm, C., Schmitt, M., Do, H.H., Rahm, E.: QuickMig: automatic schema matching for data migration projects. In: CIKM'07 Proceedings (2007)
18. Li, J., Tang, J., Li, Y., Luo, Q.: RiMOM: a dynamic multistrategy ontology alignment framework. TKDE **21**(8), 1218–1232 (2009)
19. Hu, W., Qu, Y.: Falcon-AO: a practical ontology matching system. Web Semant. **6**(3), 237–239 (2008)
20. Suchanek, F.M., Abiteboul, S., Senellart, P.: Paris: probabilistic alignment of relations, instances, and schema. In: Proceedings of the VLDB Endowment, vol. 5(3), pp. 157–168 (2011)
21. Saleem, K., Bellahsene, Z., Hunt, E.: PORSCHE: performance oriented schema mediation. Inf. Syst. **33**, 637–657 (2008)
22. Raunich, S., Rahm, E.: ATOM: automatic target-driven ontology merging. In: ICDE Proceedings, pp. 1276–1279 (2011)
23. Castano, S., Antonellis, V.D., Vimercati, S.D.C.D., Melchiori, M.: An xml-based integration scheme for web datasources. Ingénierie des Systèmes d'Information **6**(1), 99–122 (2001)
24. Dragut, E.C., Wu, W., Sistla, A.P., Yu, C.T., Meng, W.: Merging Source Query Interfaces on Web Databases. In: ICDE Proceedings, vol. 46 (2006)
25. Beneventano, D., Bergamaschi, S., Guerra, F., Vincini, M.: The MOMIS approach to information integration. In: ICEIS 2001 Proceedings, Setubal, Portugal, vol. 1, pp. 194–198 July 2001
26. Pfeifer, K., Meinecke, J.: Identifying the truth - aggregation of named entity extraction results. In: iiWAS'13 Proceedings, ACM, Austria (2013)