Mapping Text Mining Taxonomies

Katja Pfeifer and Eric Peukert
SAP AG, Chemnitzer Str. 48, 01187 Dresden, Germany

Keywords: Instance-based Matching, Text Mining, Taxonomy Alignment.

Abstract: Huge amounts of textual information relevant for market analysis, trending or product monitoring can be found on the Web. To make use of that information a number of text mining services were proposed that extract and categorize entities from given text. Such services have individual strengths and weaknesses so that merging results from multiple services can improve quality.
To merge results, mappings between service taxonomies are needed since different taxonomies are used for categorizing extracted information. The mappings can potentially be computed by using ontology matching systems. However, the available meta data within most taxonomies is weak so that ontology matching systems currently return insufficient results.
In this paper we propose a novel approach to enrich service taxonomies with instance information which is crucial for finding mappings. Based on the found instances we present a novel instance-based matching technique and metric that allows us to automatically identify equal, hierarchical and associative mappings. These mappings can be used for merging results of multiple extraction services. We broadly evaluate our matching approach on real world service taxonomies and compare to state-of-the-art approaches.

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 (Grimes, 2008). 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 Hotho et al., 2005). These techniques include the classification of text documents, the recognition of entities and relationships as well as the identification of sentiments. Recently, many of these text mining techniques were made publicly available as Web Services (e.g. OpenCalais, 2013; AlchemyAPI, 2013) to simplify their consumption and application integration. Individual services often have specific strengths and weaknesses. By combining them the overall extraction quality and amount of supported features can be increased (Seidler and Schill, 2011).

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 Figure 1. Entities have been annotated by several text mining services (OpenCalais, 2013; Evri, 2012; AlchemyAPI, 2013; FISE, 2013) that rely on different taxonomies to annotate found entities. For instance the text sequence *Airbus* is annotated with three 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 is required. Finding mappings between different service taxonomies manually is not feasible as the taxonomies can be very large and evolve over time (e.g., AlchemyAPI uses

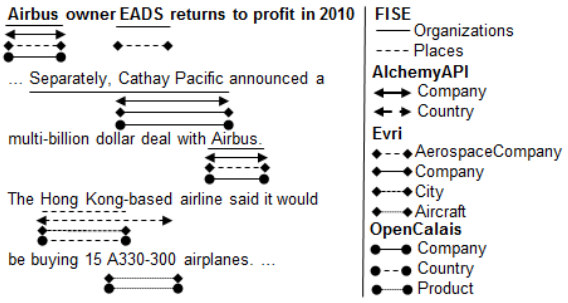


Figure 1: Analysis of a business news by several named entity recognition services (retrieved on March 9, 2011).

Pfeifer K. and Peukert E..
Mapping Text Mining Taxonomies.
DOI: 10.5220/0004500400050016
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval and the International Conference on Knowledge Management and Information Sharing* (KDIR-2013), pages 5-16
ISBN: 978-989-8565-75-4

a taxonomy with more than 400 entity types). Unfortunately applying existing (semi-)automatic ontology and schema matching techniques (Euzenat and Shvaiko, 2007; Rahm and Bernstein, 2001) 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 just associated, 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 paper:

- We introduce a novel approach of using instance enrichment to support taxonomy matching. A basic enrichment algorithm is used to populate taxonomies of text mining services with instance data by running the services on sample documents and collecting the produced annotations.
- Based on these instances a new taxonomy alignment approach is presented that uses a combined matching strategy.
- In particular, a novel metric for instance-based matchers is proposed that is able to identify equal, hierarchical and associative mappings. The metric is generic and could well be applied for other instance-based matching tasks.
- The application of the taxonomy alignment process is broadly evaluated on a number of real-world text mining services and their taxonomies. For that purpose reference mappings were created through an online survey with numerous participants. We compare to state-of-the-art instance-based alignment methods that are used in ontology matching systems.

The remainder of the paper is structured as follows: In Section 2 we formally describe the problem and introduce the notation being used within this paper. Section 3 introduces our taxonomy alignment process and presents the instance enrichment algorithm, 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 Sections 4 and 5. We introduce an exemplary application that makes use of the introduced taxonomy alignment process in Section 6 before we review related work in Section 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. In order to enable the aggregation of results of various text mining services a mapping between the different underlying taxonomies is required. However, finding such a mapping is challenging even though the names of the taxonomy types being presented to the user when annotating text are typically clear and easy to understand. 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 paper we want to apply ontology- and schema matching techniques (Euzenat and Shvaiko, 2007; Rahm and Bernstein, 2001) 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 Section 7 for a complete review of related work). Furthermore, no instances exist for most of the text mining taxonomies.

To overcome the aforementioned limitations, we propose an instance enrichment algorithm that populates the taxonomy types with meaningful instances.

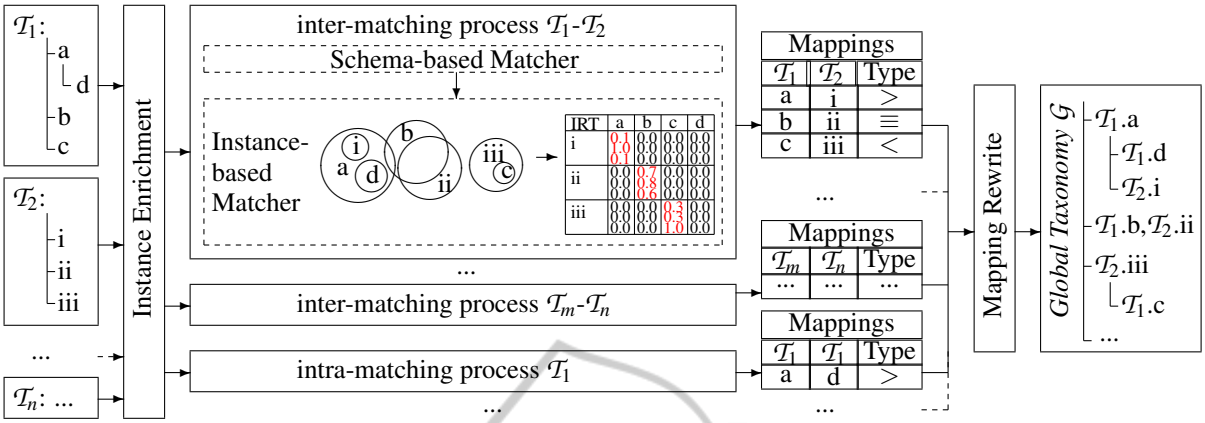


Figure 2: taxonomy alignment process.

This allows us to apply instance-based matchers and smilarity metrics like Jaccard and Dice (Isaac et al., 2007; Massmann and Rahm, 2008) 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 $\left|I(T_{ij})\right|$. 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. The process consists of several new techniques such as the instance enrichment algorithm, the intersection ratio triple (IRT) metric and several enhancements of the matching process that are presented in detail in Section 3.2 to 3.4.

3.1 Overall Alignment Process

The general taxonomy alignment process is depicted in Figure 2. The overall idea is to retrieve mappings for the taxonomy types by a matching process. Based on the mappings a global taxonomy $\mathcal{G}$ is derived. 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). Before the mappings are integrated they 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 a mapping rewrite step as done by existing ontology matching tools like ASMOV (Jean-Mary et al., 2009). However, this step is beyond the scope of this paper and will be described in future work. 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 Figure 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 we present in Section 3.2. Furthermore the instance-based matcher applies a novel 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 Section 3.3 and give details on the inter- and intra-matching process in Section 3.4.

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 Instance Enrichment Algorithm

Usually, no instances are directly available for text mining taxonomies. To follow an instance-based matching approach as proposed in Section 3.1 the taxonomy needs to be enriched with instance data (if complete sets of instances are already available for all services the instance enrichment step can be omitted). In the following we propose an instance enrichment algorithm applicable for named entity recognition (NER) services and their taxonomies. However, the general process can be transferred to other text mining services and their taxonomies.

Instances of an entity type can be obtained by executing the services on text documents and collecting the extracted information. Depending on the service, concrete text instances (e.g., the text snippet *Barack Obama*) can be assigned to several entity types (e.g. *Person*, *Politician*, *USPresident*) or to only one of those types (e.g. *USPresident* as it is the narrowest entity type).

The general idea of the instance enrichment algorithm is to enter a number of text documents into each of the text mining services whose taxonomies are to be matched. The NER results of the services (i.e., the text snippets with assigned categories) are grouped by each entity type for each service. In order to consider the entity disambiguation feature (e.g., *Paris* is recognized as *City* and in another context as *Person*) the context of the entity instances (i.e., document name and position) is attached to the text snippet.

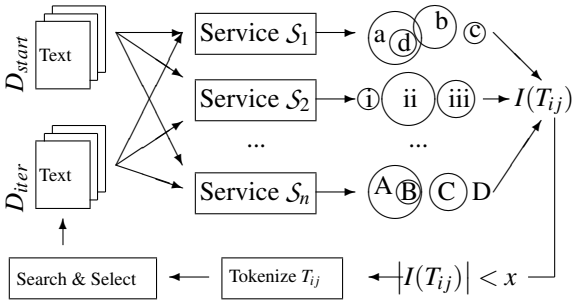


Figure 3: general instance enrichment process.

It is obvious that the generated instances are directly depending on the document set and the quality of the text mining services. We observed that only a subset of the entity types from the extraction taxonomies we took into account were enriched with instances when taking arbitrary text documents. For that reason we propose an iterative instance enrichment algorithm for the taxonomies $\mathcal{T}_i$ of the considered services $\mathcal{S}_i (i = 1, 2, ..., n)$ as follows (see Figure 3 for an illustration of the iterative process):

1. Randomly select a fixed number of documents D_{start} from a document base that covers a huge amount of different concepts (e.g., articles from Wikipedia). Set $D = D_{start}$, $iter = 0$ and create empty instance sets $I(T_{ij})$ for each element $T_{ij} \in \mathcal{T}_i$ for each of the taxonomies $\mathcal{T}_i$.
2. Enter the documents D into the text mining services $\mathcal{S}_i$ and cluster the results on the entity types $T_{ij} \in \mathcal{T}_i$ for each taxonomy. Add the retrieved instances into the instance sets $I(T_{ij})$.
3. Select the entity types T_{ij} without any instances (optionally: with less than x instances) in the instance sets $I(T_{ij})$ (i.e., $|I(T_{ij})| = 0$ or $< x$). If the number of those entity types is zero stop the iteration, else tokenize the names of these entity types.
4. Search the document base by using the particular extracted tokens as search string (e.g., search Wikipedia). Take the f first results of this search not yet having been included in D and add these documents to D_{iter}.
5. Set $D = D_{iter}$, increment $iter$ and go on with step 2. Iterate as long as the fixed maximum number of iterations $iter_{max}$ is reached or step 3 aborts the process.

The process for the generation of a qualified docu-

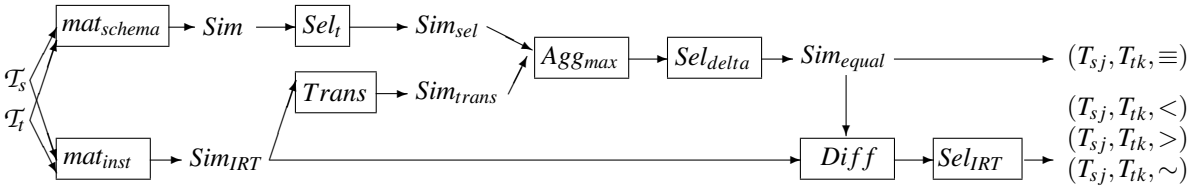


Figure 4: Combined Matching Strategy.

ment set described here can be automatically executed and therefore fits perfectly for a self-acting matching process. With our adaptive approach we are able to retrieve a high number of instances with only few service calls. This is important since calling services takes time and is costly.

3.3 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 Isaac et al. (2007):

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

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 coefficient considers the frequency of co-occurring instances with its correction factor c. It assigns smaller similarity scores to element pairs whose instances co-occur less frequently. That means, that a smaller score is assigned to one co-occurring instance in a union set of two instances compared to 100 co-occurring instances in a 200 instances large union set (the classical Jaccard coefficient would assign 0.5 to both cases). For details how to configure c please refer to Isaac et al. (2007).

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 NER 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:

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

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

Figure 5 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 Figure 1).

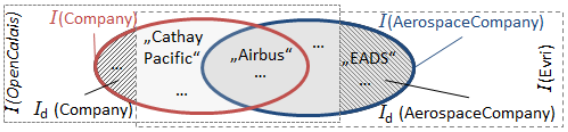


Figure 5: Example for quality restrictions.

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.

3.4 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 Figure 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 in COMA++ (Do and Rahm, 2002). 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 Figure 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 Section 3.3 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:

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

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

with $\lambda > 0$ and $0 \le z \le 1$

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 Do and Rahm, 2002). We apply the MaxDelta selection from Do and Rahm (2002) 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 Section 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 (2013), AlchemyAPI (2013) and Evri (2012). We only considered the taxonomies that are provided for English text. The entity taxonomy of OpenCalais is documented on the service website and in an OWL ontology. It 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). The *Type*-attributes allow to derive entity subtypes (e.g., Person_Sports, Person_Entertainment). All in all the OpenCalais taxonomy consist of 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 (by first retrieving facets and then requesting the entities and subentities for these facets). 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 exist. In order to evaluate the quality of the mappings retrieved with our approach, we manually produced a gold standard. To minimize the matching problem we sampled the more than 180,000 entity type pairs by selecting only those entity pairs, for which the generated instances overlapped (i.e., both entity types had at least one instance in common). The remaining roughly 4,500 entity type pairs were used for human evaluation. We assume that the influence of sampling the entity type pairs is marginal – if there was no overlap of the instance sets retrieved by our instance enrichment algorithm it is unlikely that there will be any overlap of the instances and a potential relation between the taxonomy types when using the services on arbitrary text documents.

In an online evaluation the entity type pairs plus some sample instances and links to the taxonomies were presented to approximately 40 people. They had to assign the relations "equivalent to", "broader than", "narrower than", "related to" and "no link" to each of the pairs (if unsure they were able to skip the decision) as long as they liked to go on. The online evaluation was run as long as a minimum of two ratings per entity type pair were retrieved. All entity pairs with different ratings were manually checked and a decision for the best rating in consideration of the two entity types was taken (that had been the case for around 1000 entity pairs). The gold standard was further refined when wrong/missing gold standard mappings were identified during the evaluation phase. Overall the imprecision of the information retrieved by the online evaluation was surprisingly high and again indicated that a manual matching and integration of the text mining taxonomies is not feasible.

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 Section 3.4. 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.

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 (Chua and Kim, 2012) – 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 Section 3.4 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 with the evaluation of the instance enrichment algorithm in Section 5.1 to show that the iterative process can be applied to retrieve a meaningful set of instances. Afterwards we compare the IRT metric to state-of-the-art metrics for instance-based matching in Section 5.2. Finally, we rate the performance of the overall intra- and inter-matching processes in Section 5.3.

5.1 Instance Enrichment

First of all we evaluated our instance enrichment process presented in Section 3.2. We used the English Wikipedia articles as a document base. Furthermore,

we set $iter_{max} = f = 1$ and slightly adapted the iteration process as follows: (1) The start documents were selected from articles of the Wikipedia category *Featured articles*[1]. (2) We split step 4 into step 4a doing the search with the extracted token as search string and 4b doing the search among Wikipedia lists. This extension was implemented as it is more likely to find instances for the entity type in a list connected to it (and possibly including an enumeration of instances) than in an article about it.

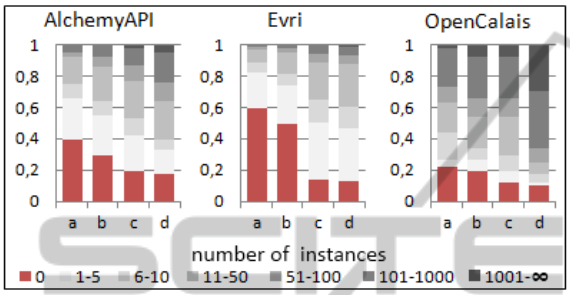


Figure 6: Instance enrichment with a) 50 randomly selected featured Wikipedia articles, b) extension to 100 articles, c) extension with Wikipedia search of respective tokenized entity types without instances, d) extension with search for the other services.

The gains we made with respect to the coverage rate of the entities when applying the instance enrichment algorithm are outlined in Figure 6. One can see that the focused extension of the document set improved the coverage rate significantly. However, we observe that even after this extensive process a lot of entity types remain with no or few instances. Although it seems to be a trivial process to generate instances for the entity types it is more complex than expected. Reasons for this are: the partly very complex entity type taxonomies, the occurrence of very specific types in the hierarchies and apparently also the inability of the services to extract all the types of their taxonomy.

5.2 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 Section 4.3 for the matcher configurations). The results of the comparison are depicted in Figure 7, 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.

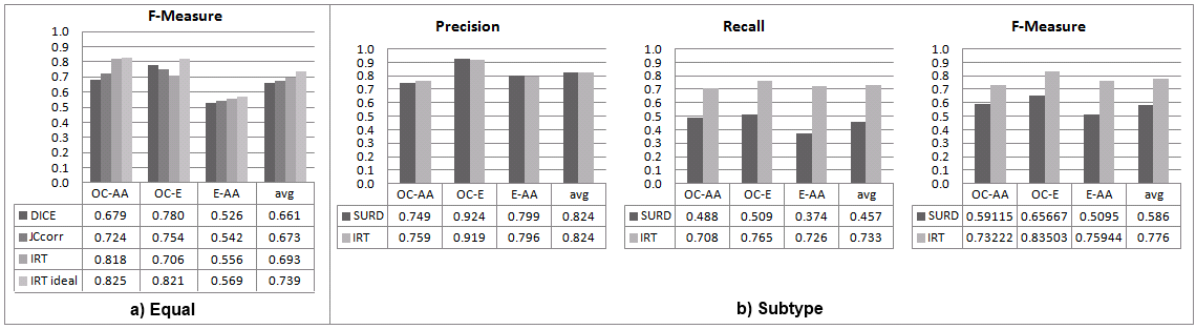
Figure 7 a) shows the F-Measure for retrieving equality mappings. We were able to slightly increase the average F-measure compared to the classical metrics Dice and corrected Jaccard. When individually setting the threshold (e.g., by using the schema-based matcher as indicator) the F-measure as well as precision 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 7 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.

5.3 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. The performance results are given in Figure 8. We compared the mapping results of the intra-matcher to the relations given within the taxonomy structure. 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

[1]Featured articles are articles that are considered to be the best in Wikipedia. At the time of our evaluation there had been 3,269 featured articles in the English Wikipedia.



a) Equal	OC-AA	OC-E	E-AA	avg
DICE	0.679	0.780	0.526	0.661
JCcorr	0.724	0.754	0.542	0.673
IRT	0.818	0.706	0.556	0.693
IRT ideal	0.825	0.821	0.569	0.739

b) Subtype	OC-AA	OC-E	E-AA	avg
SURD	0.749	0.924	0.799	0.824
IRT	0.759	0.919	0.796	0.824

b) Subtype	OC-AA	OC-E	E-AA	avg
SURD	0.488	0.509	0.374	0.457
IRT	0.708	0.765	0.726	0.733

b) Subtype	OC-AA	OC-E	E-AA	avg
SURD	0.59115	0.65667	0.5095	0.586
IRT	0.73222	0.83503	0.75944	0.776

Figure 7: Comparison of similarity metrics.

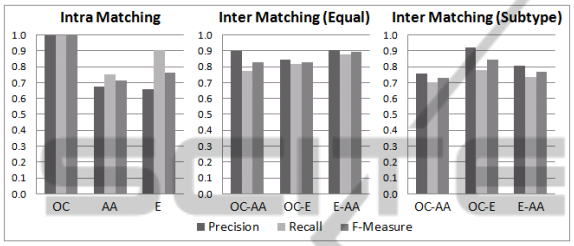


Figure 8: Performance of our matching approach.

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 been raised by more than 15% compared to the instance-based matcher only approach (see Figure 7). An average F-measure of 85% for equal and 77% for subtype 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 APPLICATION OF TAXONOMY ALIGNMENT

In order to illustrate the value of computing taxonomy alignments between extraction services we implemented a web news analysis application. We show that the following issues can be solved by combining multiple services and their annotation results: (1) The number of identified entities per category is often very small for a single service. Merging result from multiple services could increase the number of entities per category. (2) By combining taxonomies existing categorizations can be refined. (3) Individual strength of services are combined.

The computed mappings between OpenCalais and AlchemyAPI retrieved within our experimental evaluation (see Section 5.3) are taken and an integrated taxonomy/graph of categories is automatically constructed. A small subset of entity types from both services and the merged taxonomy is shown in Figure 9. The merged taxonomy consists of categories from AlchemyAPI(AA) and OpenCalais(OC). For equal matches the categories were merged. The merged taxonomy brings structure to flat lists of categories. For instance in AA *Disease*, *CauseOfDeath*, *MedicalCondition* were in no special relation. In the merged taxonomy, *Disease* and *CauseOfDeath* are now subtypes of *MedicalCondition*.

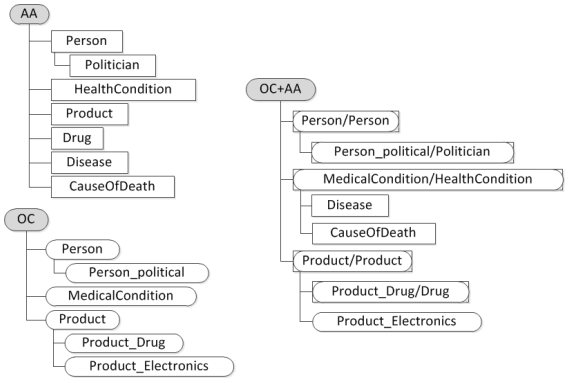


Figure 9: Merging Taxonomies AA and OC.

By using both services we analyzed web news entries from Reuters Top-News Archives from September and October 2012. All news are annotated and found entities are collected by day. The number of found entities per day can be visualized as sparkline diagrams (see Figure 10) which help to identify interesting, possible hot topics. Peaks point to days where a specific type of entity was identified particularly

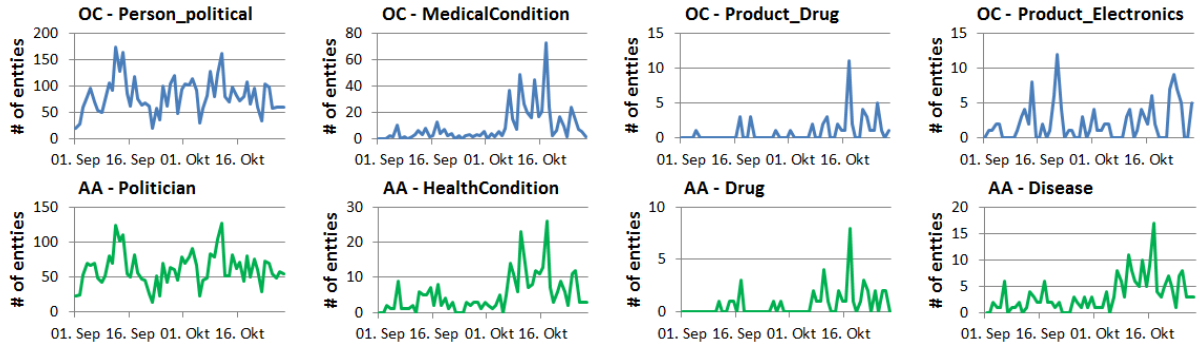


Figure 10: Sparkline diagramms.

often. The automatic taxonomy alignment process identified *Person_political* (OC) and *Polititian* (AA) as well as *MedicalCondition* (OC) and *HealthCondition* (AA) as equal matches. That these computed mappings are correct can now be verified quite well by the similarity of the generated sparklines of entity frequencies per day.

Some entity types can only found by one of the services like *Product_Electronics* (OC). The first peak on September 21st relates to the Iphone 5 release whereas on October 24th Apples "Ipad mini" was presented. Both events could not be observed with AlchemyAPI due to the missing category. Thus the combination of both services is reasonable. For the politicians, both sparklines have similar peaks, one on September 12 and October 12. For instance, on September 12 or shortly before, an attack on the US Embassy in Benghazi (Lybia) took place. Many news articles referred to comments that were given by politicians which led to a peak for politicians. Similar peaks can be observed with *MedialCondition* and *HealthCondition*. These peaks can be explained by a severe outbreak of meningitis in the US which caused a number of deaths. The reason for that outbreak was a drug that was used for patients with back pain. This also resulted in a peak for *Drugs* (AA) accordingly. Still, the number of identified entities for *Product_Drug* (OC), Drug (AA) are very low. Merging found entities from both services would increase data quality and therefore also other events of smaller scale could possibly automatically be identified.

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 Shvaiko and Euzenat, 2005; Rahm and Bernstein, 2001). 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 (Drumm et al., 2007), COMA++ (Do and Rahm, 2002), RiMOM (Li et al., 2009) or Falcon (Hu and Qu, 2008) 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 (Drumm et al., 2007; Hu and Qu, 2008; Li et al., 2009), others employ metrics that rely on the overlap of instance sets (Do and Rahm, 2002). The latter rely on similarity metrics like Jaccard, corrected Jaccard, Pointwise Mutual Information, Log-Likelihood ratio and Information Gain (see Isaac et al., 2007). Massmann and Rahm (2008) 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 (Suchanek et al., 2011) 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. Chua and Kim (2012) 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 heuris-

tics this metric is more inaccurate than the IRT metric presented in this paper. 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.

Our instance enrichment approach is crucial since it allows us to apply instance-based matching techniques in the first place. Closest to that idea is the QuickMig system (Drumm et al., 2007) where instances have to be provided manually in a questionnaire. None of the existing systems is able to generate instances beforehand to apply instance matching as we do in this paper. Moreover, we are the first to apply ontology matching techniques for matching text mining taxonomies.

8 CONCLUSIONS AND FUTURE WORK

In this paper 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 developed an instance enrichment algorithm that allows us to apply instance matching techniques in a complex matching strategy. 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 Seidler and Schill (2011) 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.

REFERENCES

AlchemyAPI (2013). AlchemyAPI Homepage. http://www.alchemyapi.com/. March 2013.

Chua, W. W. K. and Kim, J.-J. (2012). Discovering Cross-Ontology Subsumption Relationships by Using Ontological Annotations on Biomedical Literature. In *ICBO*, volume 897 of *CEUR Workshop Proc.*

Do, H. H. and Rahm, E. (2002). COMA - A System for Flexible Combination of Schema Matching Approach. In *VLDB Proc.*

Drumm, C., Schmitt, M., Do, H.-H., and Rahm, E. (2007). QuickMig: Automatic Schema Matching for Data Migration Projects. In *CIKM'07 Proc.*

Euzenat, J. and Shvaiko, P. (2007). *Ontology Matching*. Springer-Verlag.

Evri (2012). Evri Developer Homepage. http://www.evri.com/developer/. June 2012.

FISE (2013). Furtwangen IKS Semantic Engine project page. http://wiki.iks-project.eu/index.php/FISE. March 2013.

Grimes, S. (2008). Unstructured data and the 80 percent rule. http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/. Clarabridge Bridgepoints.

Hotho, A., Nürnberger, A., and Paaß, G. (2005). A Brief Survey of Text Mining. *LDV Forum*, 20(1):19–62.

Hu, W. and Qu, Y. (2008). Falcon-AO: A practical Ontology Matching System. *Web Semantics*, 6(3):237–239.

Isaac, A., Van Der Meij, L., Schlobach, S., and Wang, S. (2007). An Empirical Study of Instance-Based Ontology Matching. In *ISWC'07 Proc.*, pages 253–266.

Jean-Mary, Y. R., Shironoshita, E. P., and Kabuka, M. R. (2009). Ontology Matching with Semantic Verification. *Web Semantics*, 7(3):235–251.

Li, J., Tang, J., Li, Y., and Luo, Q. (2009). RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *TKDE*, 21(8):1218–1232.

Massmann, S. and Rahm, E. (2008). Evaluating Instance-based Matching of Web Directories. In *WebDB'08 Proc.*

OpenCalais (2013). Calais Homepage. http://www.opencalais.com/. March 2013.

Rahm, E. and Bernstein, P. A. (2001). A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal*, 10:334–350.

Seidler, K. and Schill, A. (2011). Service-oriented Information Extraction. In *Joint EDBT/ICDT Ph.D. Workshop'11 Proc.*, pages 25–31.

Shvaiko, P. and Euzenat, J. (2005). A Survey of Schema-Based Matching Approaches. *Journal on Data Semantics IV*.

Suchanek, F. M., Abiteboul, S., and Senellart, P. (2011). Paris: probabilistic alignment of relations, instances, and schema. *Proc. VLDB Endow.*, 5(3):157–168.