A Machine Learning Approach to Multilingual and Cross-lingual Ontology Matching

Dennis Spohr[1], Laura Hollink[2], and Philipp Cimiano[1]

[1] Semantic Computing Group, CITEC, University of Bielefeld
[2] Web Information Systems Group, Delft University of Technology

Abstract. Ontology matching is a task that has attracted considerable attention in recent years. With very few exceptions, however, research in ontology matching has focussed primarily on the development of monolingual matching algorithms. As more and more resources become available in more than one language, novel algorithms are required which are capable of matching ontologies which share more than one language, or ontologies which are multilingual but do not share any languages. In this paper, we discuss several approaches to learning a matching function between two ontologies using a small set of manually aligned concepts, and evaluate them on different pairs of financial accounting standards, showing that multilingual information can indeed improve the matching quality, even in cross-lingual scenarios. In addition to this, as current research on ontology matching does not make a satisfactory distinction between multilingual and cross-lingual ontology matching, we provide precise definitions of these terms in relation to monolingual ontology matching, and quantify their effects on different matching algorithms.

Keywords: Multilingual and cross-lingual ontology matching, machine learning, interoperability of financial information

1 Introduction

Ontology matching is a discipline that has matured considerably over the last years, which is shown by the fact that many ontology matching algorithms have been successfully implemented and evaluated.[3] However, while these algorithms generally focus on using information in a single language that is shared by two ontologies, *multilingual* and *cross-lingual* aspects of ontology matching are – with the notable exception of [3, 10] – still largely understudied. Assuming that more and more ontological resources will become available on the Semantic Web in more than one language, it is necessary to develop novel algorithms which are able to leverage multilingual information in case it is available, as well as capable of bridging the gap in case two ontological resources which do not share any languages need to be matched.

[3] See e.g. `http://oaei.ontologymatching.org/`.

In this paper, we present a new approach that relies on machine learning techniques in order to match concepts in two ontologies using both multilingual and cross-lingual information. As a very challenging use case, we have chosen financial accounting standards (FAS) such as the *United States Generally Accepted Accounting Principles* (US-GAAP) or the German *Grundsätze ordnungsmäßiger Buchführung* (GoB) – not only because they represent a type of taxonomic resource that raises a number of methodological issues, but also because the lack of interoperability of financial information across jurisdictional barriers is one of the most central problems in the business domain.[4] On the one hand, this is because companies from different countries are required to report their financial statements against different FAS. As these use different financial concepts with different interpretations, financial data reported against e.g. US-GAAP cannot be compared to financial data based on GoB, as the concepts need to be matched before any meaningful data integration can take place. On the other hand, FAS are frequently multilingual (i.e. annotated with more than one language) and have more than one label per language, which raises the question as to how to match financial concepts from ontologies sharing more than one language, or concepts from ontologies which are multilingual but do not share any languages. FAS thus represent one of the primary obstacles to achieving interoperability, and the impact of an approach that helps to solve this problem is expected to be considerably high.

This paper addresses the afore-mentioned issues in several respects. Firstly, we will give precise definitions of multilingual and cross-lingual ontology matching in relation to monolingual ontology matching, and discuss general research questions arising in such settings (Section 2). Moreover, we discuss several approaches which leverage multilingual and cross-lingual information in order to learn a matching function between two ontologies, using a small set of manually aligned financial concepts as training data. In contrast to the predominant view of ontology matching as a *classification problem* (as e.g. in [4, 9]), we understand it as a *ranking problem*, similar to relevance ranking in information retrieval. In particular, we describe a novel approach that trains a *ranking support vector machine* (see [5]) on relative preference constraints between a concept in a source ontology and all possible concepts in a target ontology, with the goal of ranking good matches higher than bad matches. The approach is described in detail in Section 3 and evaluated on different pairs of FAS in different scenarios, in order to quantify the impact of multilingual and cross-lingual information on the performance of ontology matching algorithms (see Section 4).

2 Background and preliminaries

2.1 Monolingual, multilingual and cross-lingual ontology matching

Current research on ontology matching does not make a consistent distinction between *multilingual* and *cross-lingual* ontology matching (see e.g. [3, 6, 10]). In

[4] See `http://www.xbrl.org/2010TechDiscussion/2010TechDiscussion.pdf`.

the following, we will define these notions, based on the general definition of ontology matching as "the process of finding relationships or correspondences between entities of different ontologies" (cf. [2]). As the definitions focus on those aspects of ontology matching which are relevant to multilingual and cross-lingual scenarios, specific strategies – such as structure-based or instance-based matching – will be ignored. Given a source ontology S and a target ontology T, the sets $S(l)$ and $T(l)$ of labels of S and T in a language l, and the sets L_S and L_T of languages in S and T respectively, these notions can be defined as follows.

Definition 1. ***Monolingual ontology matching*** *is the process of matching entities in S and T by comparing the labels in $S(l)$ and $T(l)$ in a single language $l \in L_S \cap L_T$.*

Definition 2. ***Multilingual ontology matching*** *is the process of matching entities in S and T by comparing the labels in $S(l_i)$ and $T(l_i)$ in at least two languages $l_i \in L_S \cap L_T$, with $|L_S \cap L_T| \geq 2$.*

Definition 3. ***Cross-lingual ontology matching*** *is the process of matching entities in S and T either*

a. by translating the labels in $S(l)$ to at least one language $l' \in L_T$ and comparing the labels in $S(l')$ with those in $T(l')$, or
b. by translating the labels in $T(l)$ to at least one language $l' \in L_S$ and comparing the labels in $S(l')$ with those in $T(l')$, or
c. by translating the labels $S(l)$ and the labels $T(l')$ to at least one language $l'' \notin L_S \cup L_T$ and comparing the labels in $S(l'')$ with those in $T(l'')$.

For example, given a source ontology S with labels in English, German and Italian, monolingual ontology matching is a process that matches entities in S to entities in a target ontology T_1 with English labels by comparing the English labels in S with those of T_1. Multilingual ontology matching is a process that matches entities in S with entities in a target ontology T_2 with English and German labels by considering the labels in English and German. Cross-lingual ontology matching is a process that matches entities in S to entities in a target ontology T_3 with French labels either by translating the labels of S to French (Definition 3a.), by translating the labels of the T_3 to one of the languages in L_S (Definition 3b.), or by translating the labels of S and T to a third language (Definition 3c.). We believe that e.g. what Fu et al. [3] refer to as "multilingual ontologies" can thus be described more accurately as a cross-lingual matching scenario involving two (or more) monolingual ontologies.

2.2 Financial accounting standards and XBRL

As was mentioned in the introduction, financial accounting standards (FAS) differ between countries, and thus inhibit interoperability of financial information. However, there have been important developments towards solving this problem in recent years. From a technological perspective, the *eXtensible Business Reporting Language* (XBRL; [11]) solves the syntactic aspects of this interoperability issue by providing a common XML-based framework for expressing financial information. In XBRL, a FAS is commonly referred to as a *taxonomy*, as it specifies – among others – a hierachical structure according to which financial concepts appear in a financial statement (called *presentation hierarchy* in XBRL). In addition to this, for financial concepts which determine monetary values, such taxonomies specify how the value in question is to be calculated (e.g. *"Assets"* is the sum of *"Current assets"* and *"Non-current assets"*). Similar to the hierarchical presentation structure, these calculations are recursive (e.g. *"Non-current assets"* is, in turn, the sum of *"Property, plant and equipment"*, *"Investment property"* etc.). This means that a monetary concept has a number of calculation items, each of which may itself be calculated on the basis of further calculation items. As such, we can distinguish between the *direct calculation items* of a monetary concept (*"Current assets"* and *"Non-current assets"* for *"Assets"*), as well as the *elementary calculation items* (e.g. *"Investment property"*), and likewise between *direct* and *elementary children* in the case of the presentation hierarchy. Finally, a concept can have more than one calculation, and more than one presentation.

While the move towards XBRL has been a crucial development towards achieving interoperability, it constitutes only a first step. In particular, it does not solve the conceptual aspects of the problem, as companies from different countries still use different vocabularies to file their financial reports. As a result, the semantics of individual pieces of information is still not interchangeable. The XBRL Europe Business Registers Working Group (XEBR WG) has tried to approach this problem from a conceptual perspective, by defining a set of core financial concepts which are believed to be shared by most FAS. The main idea behind this activity is that if the taxonomy of core financial concepts defines exact and close matches to the different national FAS, financial information reported against each individual FAS becomes interoperable through these mappings. While the work of the XEBR WG is still ongoing, first manual matches between the XEBR core taxonomy and several national FAS have already been produced, and can thus be leveraged for the approach described in this paper. Moreover, a very beneficial side-effect of the resource created by the XEBR WG is that it is possible to define matches between the individual taxonomies as well, based on the manual matches to the core taxonomy. Since – as was mentioned in the introduction – these taxonomies are frequently annotated in more than one language, the XEBR WG has created a valuable resource for the investigation and evaluation of different multilingual and cross-lingual matching strategies.

2.3 Open research questions in ontology matching

Trojahn et al. [10] mention that multilingual and cross-lingual ontology matching is an open research issue. In this section, we try to explicate some of the research questions arising in such scenarios, with a particular focus on the machine learning aspect.

Impact of machine translation in cross-lingual scenarios. Fu et al. [3] have argued that the matching quality in cross-lingual scenarios strongly depends on the translation quality of label translations generated by machine translation (MT) tools. This certainly holds for the present study as well, since the choice of the MT system determines e.g. whether the Italian term "Conto economico" is translated as "Income statement"[5] or as "I count economical"[6]. The conclusion of Fu et al. [3] is that good translation quality is a prerequisite for achieving good quality cross-lingual ontology matches. While we do believe that this is true when comparing cross-lingual ontology matching to monolingual ontology matching with high-quality labels, it is worth investigating the possibility of translating the labels of both ontologies to a third language, as it may be the case that the quality difference between the labels can thus be reduced. In addition to this, in a machine learning scenario, a learning algorithm may weight structure-based similarity features higher in case string-based ones are found to be less predictive or even unpredictive, thus reducing the importance of high-quality translations.

Impact of structural information in ontology matching. Previous work has already shown the importance of structural information in ontology matching (see e.g. [2]). However, while it seems to be intuitively the case that algorithms capable of leveraging structural information should perform better than those which do not have this kind of information available, the question in a machine learning scenario is whether a learning algorithm which does not have access to structural information can still learn a reasonably predictive matching function. Therefore, a direct comparison between an algorithm using structural information with one not using structural information is necessary in order to answer this question.

Aggregation of scores in multilingual scenarios. Matching concepts with annotations in several languages, as well as several annotations within a single language, raises a number of further questions. One of these is the question how the similarity scores across different annotations should be aggregated within a single language (*intralingual aggregation*) as well as between languages (*interlingual aggregation*; cf. "composition" in [6]). For example, should the fact that one label of a concept C_S in a language l is very similar to one label of a concept C_T in language l suffice to say that C_S and C_T are good matching candidates? Or is the average over all labels within a language – averaged over all languages – a better indicator? To illustrate the importance of the treatment of multilingual information in ontology matching, consider the following example. The XBRL taxonomy of the *International Financial Reporting Standards* 2009 specifies a label "Total property, plant and equipment, gross" for the respective concept `PropertyPlantAndEquipment`. In the Italian GAAP, the corresponding concept is called "Total tangible fixed assets". Comparing only these two labels in a single language would yield a very low similarity score, as the only overlap consists in the word "total". However, both taxonomies specify labels in Italian and French. While the overlap in the Italian labels is still only marginally higher than in the English ones ("Immobili, impianti e macchinari" vs. "Immobilizzazioni materiali"), the French set of labels assures that the two concepts are in fact equivalent

[5] Using Microsoft's MT system Bing; `http://www.microsofttranslator.com/`.
[6] Using SDL FreeTranslation; `http://www.freetranslation.com/`.

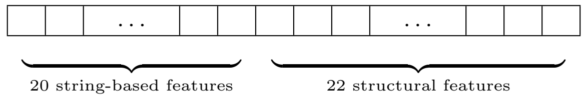

Fig. 1. Vector containing 42 features measuring the similarity between two concepts

("Immobilisations corporelles" vs. "II Immobilisations corporelles"). This example not only shows how vital multilingual information is for ontology matching, but also that different strategies for intralingual and interlingual matching need to be defined. In the following section, we present the main ideas of our approach, as well as the features implementing the different strategies.

3 Machine learning approach to ontology matching

As was mentioned above, the general idea of our approach is to apply machine learning techniques to ontology matching, based on the notion of *ranking* SVM*s* as defined by Joachims [5]. In order to be able to apply this methodology, we are in need of a set of manually matched concepts to train on, as well as features representing the characteristics of each possible match. For the first issue, we can resort to the work of the XBRL *Europe Business Registers Working Group* (XEBR WG), which is currently in the process of matching different FAS to a set of core concepts. For the second issue, we define an appropriate set of features such that each combination of a source concept C_S with a target concept C_T can be represented as a feature vector, as in Figure 1. As each of these specifies the similarity between C_S and C_T, the value of each feature is between 0 and 1.

In total, we have thus defined 42 different features, comprising 20 string-based features, as well as 22 structure-based ones. These will be discussed in more detail below, before describing how the algorithm can be applied to the resulting similarity vectors.

3.1 Definition of feature set

String-based features. Similar to most other approaches in the field, we make use of a number of different string-based comparisons in order to measure the similarity between to concepts. In particular, we use five different measures, each of which represents a feature in the vector in Figure 1. Two similarity features are based on the Levenshtein edit distance measure [2, 8], where one is applied to the labels of C_S and C_T, and the other one to the labels after their tokens have been sorted. This is to cover cases like "Current assets" vs. "Assets, current", where the plain (unsorted) Levenshtein distance would be very high although the labels are in fact very similar. Two further features use a bag-of-words cosine similarity measure, one on the original labels and the other one after punctuation has been removed. The fifth string-based measure uses the following substring distance as implemented by Euzenat and Shvaiko [2][7].

[7] See [2] also for a more detailed description of the different measures

(1) $sim_{substr}(label_1, label_2) = \frac{2 * |longest\ common\ sequence|}{|label_1|+|label_2|}$

As mentioned in the previous section, we distinguish between different ways of aggregating the scores within a language, as well as between languages. In particular, we consider the best and average scores of all labels within a language, as well as the best and average scores across all shared languages. Thus, in order to cover these different intra- and interlingual matching strategies, we have implemented four different aggregations for each of the above measures. For example, the intralingual string similarity between the sets of labels of C_S and C_T in a language k (i.e. $C_S(k)$ and $C_T(k)$) using the Levenshtein measure is calculated on the basis of both (2) for the average score $sim^k_{intra\sim}$ over all labels and (3) for the best score sim^k_{intra+} of all labels.

(2) $sim^k_{intra\sim}(C_S(k), C_T(k)) = \frac{1}{n * m} \sum_{i=1}^{n} \sum_{j=1}^{m} sim_{lev}(l^i_{C_S}, l^j_{C_T})$

(3) $sim^k_{intra+}(C_S(k), C_T(k)) = max(sim_{lev}(l^1_{C_S}, l^1_{C_T}), \ldots, sim_{lev}(l^n_{C_S}, l^m_{C_T}))$

Each of these is then aggregated to yield the interlingual similarity scores. For example, the interlingual score $sim_{inter\sim/\sim}$ between two concepts C_S and C_T is calculated by taking the average similarity of the average intralingual similarity scores over all n languages shared by S and T (i.e. $|L_S \cap L_T| = n$), as in (4), while the interlingual score $sim_{inter+/\sim}$ takes the best similarity score of all average intralingual scores. The scores $sim_{inter\sim/+}$ and $sim_{inter+/+}$ are calculated analogously.

(4) $sim_{inter\sim/\sim}(C_S, C_T) = \frac{1}{n} \sum_{i=1}^{n} sim^i_{intra\sim}(C_S(i), C_T(i))$

(5) $sim_{inter+/\sim}(C_S, C_T) =$
$max(sim^1_{intra\sim}(C_S(1), C_T(1)), \ldots, sim^n_{intra\sim}(C_S(n), C_T(n)))$

This way, we have arrived at a total of 20 different string-based features, i.e. five measures times four possible aggregations.

Structural features. In addition to the string-based features, we have defined a set of 22 features representing the structural similarity between two concepts. As we cannot describe all of these features in depth, we limit ourselves to describing the calculation of two types of features to which most of the other structural features belong. In particular, we discuss the use of calculation information in S and T by considering the sets of direct and elementary items in calculations of C_S and C_T (i.e. $Cal^{dir}_{C_S} \times Cal^{dir}_{C_T}$ and $Cal^{ele}_{C_S} \times Cal^{ele}_{C_T}$). The scores using presentation information by considering direct and elementary children in the presentation hierarchy are calculated analogously.

The scores for the first type of features are rather straightforward to calculate, in that the average number[8] of direct (or elementary) items in calculations of

[8] Recall from Section 2.2 that a concept can have more than one calculation and presentation.

	$item^1_{C_S}$	$item^2_{C_S}$	$item^3_{C_S}$	$item^4_{C_S}$
$item^1_{C_T}$	0.1	0.3	0.4	0.2
$item^2_{C_T}$	0.4	0.7	0.3	0.5

Table 1. Matrix of string similarities between items of calculations of C_S and C_T

a concept C_S is compared with the corresponding number in calculations of C_T. For example, if C_S has five direct calculation items and C_T has three, then $sim^{dir}_{cal\#}(C_S, C_T) = 1 - \frac{5-3}{5} = 0.6$. Similar calculations are done for the minimal and maximal number of elementary and direct calculation items. The second type of structural feature combines structural information with string-based similarity measures. In particular, we compare not the number of direct or elementary items of the calculations of C_S and C_T, but their similarities in all languages under consideration[9]. The motivation for this is that concepts whose components have similar labels are expected to be similar, even if e.g. their own labels are very different. Consider the similarity matrix of calculation items of C_S and C_T in Table 1, which shows the pairwise string similarities between all calculation items of C_S and C_T. In order to calculate a similarity value between C_S and C_T, we apply a best-first algorithm to the matrix, which yields that $item^2_{C_S}$ is aligned with $item^2_{C_T}$ (0.7) and $item^3_{C_S}$ with $item^1_{C_T}$ (0.4). As two calculation items have not been aligned, we consider both as having 0.0 similarity with items in C_T. In other words, we divide the sum of the scores by $max(|Cal^{dir}_{C_S}|, |Cal^{dir}_{C_T}|)$. This results in an overall similarity of $sim^{dir}_{cal\$}(C_S, C_T) = \frac{0.7+0.4}{4} = 0.275$ for this example. The scores for elementary calculation items are calculated similarly, as well as the scores for children in the presentation hierarchy. As was mentioned above, we have thus defined 22 structural features, arriving at a total of 42 features on which the ranking SVM algorithm can be trained.

3.2 Learning the matching function

On the basis of the features just discussed, we can now represent all combinations of concepts in S with concepts in T in terms of their similarity scores. Moreover, since we want to apply the ranking SVM algorithm as developed by [5] in order to learn the matching function, we need to specify relative relevance preferences between the possible matches. As the XEBR WG has defined exact matches as well as broad and narrow matches, we can state preferences such that exact matches of a concept C_S should be ranked higher than broad and narrow matches, which are in turn to be ranked higher than all other possible combinations of C_S with concepts in T. Therefore, we assign a target value of 3 to exact matches, 2 to broad and narrow matches, and 1 to all other combinations.[10] Given the

[9] We have used the Levenshtein score of the pair of labels which matched best overall.

[10] This means that we consider both broad and narrow matches as “close” matches, since it did not seem reasonable to assume that narrow matches should generally be ranked higher than broad matches or vice versa. See

features and similarity vectors as presented above, we can apply the ranking SVM algorithm described by Joachims [5], which produces an SVM model predicting scores for each input similarity vector. The matches can then be ranked such that those for which the model predicts higher scores are ranked above those with lower scores. In the experiments described in this paper, we have limited ourselves to learning an SVM with a linear kernel, which produces a single support vector.[11]

4 Evaluation

In order to make the impact of multilingual and cross-lingual information evident, we have defined several scenarios differing as to the type of language information they have available. As defined in Section 2, we differentiate between monolingual matching (using one overlapping language), multilingual matching (using at least two overlapping languages), and cross-lingual matching (involving the translation of at least one of the ontologies into at least one additional language). In the latter case, we further distinguish between monolingual and multilingual cross-lingual ontology matching. In addition to this, we investigate a cross-lingual transfer scenario in which the algorithm is trained on two pairs of taxonomies and evaluated on a third pair. The different scenarios are described in Section 4.2. In order to be able to quantify the contribution of other types of information to the matching process, such as the impact of structural features and the availability of close matches, we have further defined different evaluation settings differing with respect to the amount of information they can leverage. These settings are described in 4.3. Sections 4.4 and 4.5 present and analyse the results of the evaluation.

4.1 Data

Thanks to the fact that the XBRL Europe Business Registers Working Group (XEBR WG) has begun to manually match national financial accounting standards to their taxonomy of core financial concepts, we are able to evaluate our matching algorithms on the data produced by the XEBR WG. In particular, we have used version 5 of the XEBR core taxonomy (XEBR), the Italian *Tassonomia relativa ai Principi Contabili Italiani* of 2011 (also called *ITaliaCodiceCivile*; ITCC), and the GAAP taxonomy of the German *Handelsgesetzbuch* of 2011 (HGB). Table 2 lists the sizes of the taxonomies and the languages available, as well as the matches between them as defined by the XEBR WG.

Between XEBR and ITCC, the XEBR WG has defined 61 exact and 77 close matches, with 70 XEBR concepts having at least one exact or close ITCC match.

`http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html` for implementational details of the tool used to train the ranking SVM.

11 First experiments involving a radial basis function kernel have yielded significantly worse results. For this reason, and due to the efficiency drawbacks compared to linear kernels, they have been discarded so far.

Taxonomy	Languages	Concepts
XEBR	EN	269
ITCC	EN, FR, DE, IT	444
HGB	EN, DE	3,146

	XEBR	ITCC	HGB
XEBR	×	61 (77)	64 (475)
ITCC	61 (77)	×	29 (38)
HGB	64 (475)	29 (38)	×

Table 2. Taxonomies used in the evaluation, with number of exact (and close) matches

For XEBR and HGB, 64 exact and 475 close matches were defined, with 67 XEBR concepts having at least one exact or close HGB match. Matches between ITCC and HGB were not explicitly created by the XEBR WG. However, given that there are matches from each of these taxonomies to XEBR, we have applied a simple heuristic in order to arrive at a mapping between the two. If an ITCC concept and a HGB concept were marked as an exact match of the same XEBR concept, we defined them as exact matches. If an ITCC concept was marked as a close match of an XEBR concept, and this concept had an exact HGB concept match, we marked them as close matches, and vice versa. After applying this heuristic, we arrived at 29 exact and 38 close matches between ITCC and HGB, all of which were manually inspected and verified.

Regarding the structural content of the taxonomies, it needs to be said that the XEBR taxonomy does not define calculation information, but only provides presentation information. This means that structural information in the XEBR / ITCC and XEBR / HGB pairs is limited to leveraging presentation information, while in the ITCC / HGB both presentation and calculation information is available. Finally, we have used an RDF conversion of the XBRL format in which financial concepts are represented as RDF classes.

4.2 Matching scenarios

Monolingual scenario. In the monolingual matching scenario, matching is done on the basis of one overlapping language. As English is the only language present in all taxonomies, we have used it for the monolingual matching scenario.

Multilingual scenario. As ITCC and HGB are the only taxonomies in the data set sharing more than one language, the multilingual scenario could only be applied to this pair of taxonomies, using English and German labels.

Cross-lingual scenario, S translated (monolingual). As was mentioned above, we distinguish different cross-lingual scenarios. In this first scenario, we have removed the labels in S and T such that each contained labels only in one non-overlapping language, in order to simulate a cross-lingual matching problem. For the XEBR to ITCC pair, we isolated the Italian labels in T and translated the English labels in S to Italian.[12] For the pair XEBR and HGB, we isolated the German labels in T and translated the English labels in S to German, and for the pair ITCC to HGB, we translated the Italian labels in S to English.

[12] All translations were done with the Microsoft MT system Bing.

Cross-lingual scenario, S and T translated (monolingual). This second cross-lingual scenario is similar to the previous one, except for the fact that we do not translate the labels in S to a language in T, but instead translate the labels of both S and T to a pivot language. The motivation behind this is to find out whether the translation quality issues in cross-lingual scenarios can be mitigated to some extent if the quality between both sets of labels is (at least assumed to be) more similar. In this scenario, we have translated the English labels in XEBR as well as the Italian labels in ITCC to German, and performed monolingual matching. Similarly, for XEBR to HGB, we have translated the English source labels as well as the German target labels to Italian, and for the ITCC to HGB case the Italian source labels as well as the German target labels to English.

Cross-lingual scenario, S translated (multilingual). In the third cross-lingual scenario, we have translated the labels in S to at least one other language existing in T, and performed multilingual matching. In the XEBR to ITCC case, we have translated the English labels in S to German and Italian, and performed multilingual matching on English, German and Italian[13]. For XEBR to HGB, we have translated the English source labels to German and matched in English and German. For the third pair, we first removed the English, French and German labels from ITCC, translated the remaining Italian labels to English and German, and performed multilingual matching with the English and German labels in HGB.

Cross-lingual transfer learning scenario, S translated (multilingual). In the final cross-lingual scenario, we wanted to investigate whether the matching function learned on two pairs of taxonomies can be transferred to a third pair of taxonomies, using the similarity scores calculated in the previous scenario (cross-lingual multilingual). In particular, for the XEBR to ITCC pair we trained the ranking SVM on the similarity scores between concepts in XEBR and HGB, as well as between ITCC and HGB, and tested it on the scores for XEBR and ITCC. Similarly, the XEBR to HGB pair was trained on XEBR to ITCC and ITCC to HGB, and the ITCC to HGB pair was trained on XEBR to ITCC and XEBR to HGB respectively.

4.3 Settings

In each of the scenarios just described, we have evaluated three different learning settings as well as one baseline setting.[14]

AllInfo. In this setting, a matching function is trained and tested on all available information. In particular, it uses the similarity scores of exact and close matches, as well as all structural features (i.e. similarity scores based on presentation and calculation similarity). The matches in the test set are then ranked according to the score assigned by the learned matching function.

[13] Note that five entities in ITCC needed to be translated as well, as they lacked either an English or a German label. We assume this does not distort the results too much.

[14] As the baseline setting does not involve learning, the baseline for the transfer scenario is given by the score it yields in the cross-lingual multilingual setting.

NoClose. This setting is similar to *AllInfo* except for the fact that it does not use close matches for training and testing.

NoStruc. This setting is similar to *AllInfo* except for the fact that it does not use structural information for training and testing. As was mentioned in Section 4.1, in the case of XEBR/ITCC and XEBR/HGB this means that presentation information is ignored, and in the case of ITCC/HGB that presentation and calculation information is ignored.

EqWeights. This is a baseline setting where the matching function is not learned, but instead all features are assigned the same weight. In other words, the matches are ranked simply according to their average score across all features.

4.4 Results

The results presented in this section are based on the following configuration. For each of the matching settings *AllInfo*, *NoClose* and *NoStruc*, we have carried out a four-fold cross-validation (i.e. training on three folds and testing on one fold). Each source concept in the training folds contained 20 similarity vectors representing exact, close and random bad matches with concepts in the target ontology, and they are the same 20 matches for all scenarios within one pair of ontologies. In contrast to this, for each source concept in the test folds the similarity vectors for all combinations with concepts in the target ontology T is given. This means, for example, that for the taxonomy pair XEBR and HGB – with matches defined to 67 of 3,146 HGB concepts –, each validation iteration is based on roughly 51 * 20 training examples, and evaluated on roughly 16 * 3,146 test examples (cf. Section 4.1 above).

As was mentioned in Section 3, we have used the algorithm developed by Joachims [5] to train a ranking SVM with a linear kernel. The developer of the corresponding tool SVM^{rank} gives a default value for the regularisation parameter C (i.e. the trade-off between training error and margin) of 0.01 for "normal" SVMs, and defines $C_{rank} = C * n$ (where n is the number of queries, i.e. concepts in S) for ranking SVMs. Due to this dependence on the number of concepts in the source ontology, and as the number of matches provided by the XEBR WG – and thus the number of source concepts that can be used for evaluation – differs for each pair of taxonomies, C_{rank} is different for each pair of taxonomies. However, it should be noted that we have neither tried to optimize C for a given pair of taxonomies, nor tried to find the optimal set of training samples. The results are thus all based on the default value 0.01 for C, using a simple uniform random sampling method that produces acceptable results for all taxonomy pairs. We believe that this should make the results comparable.

Table 3 shows the results for the different matching settings. The column entitled "1" indicates the cases in which the matcher has ranked the exact match at rank 1 (i.e. precision), "5" indicates that the times in which the exact match was among the first 5 ranks, and analogously for column "10". As was mentioned above, the baseline for each scenario is given by a matcher that uses the

Scenario	Setting	XEBR / ITCC			XEBR / HGB			ITCC / HGB		
		1	5	10	1	5	10	1	5	10
Monolingual	$AllInfo_1$	**51.67**	**76.67**	**81.67**	45.90	**73.77**	**78.69**	**44.83**	65.52	68.97
	$NoClose_1$	**51.67**	73.33	80.00	**52.46**	**73.77**	**78.69**	41.38	65.52	68.97
	$NoStruc_1$	46.67	66.67	76.67	50.82	72.13	**78.69**	41.38	55.17	58.62
	$EqWeights_1$	41.67	63.33	78.33	**52.46**	68.85	**78.69**	41.38	**68.97**	**72.41**
Multilingual	$AllInfo_n$	–	–	–	–	–	–	**51.72**	68.97	68.97
	$NoClose_n$	–	–	–	–	–	–	**51.72**	68.97	72.41
	$NoStruc_n$	–	–	–	–	–	–	44.83	55.17	65.52
	$EqWeights_n$	–	–	–	–	–	–	44.83	**72.41**	**75.86**
Cross-lingual, S and T translated	$AllInfo_1^{ST}$	**38.33**	**63.33**	**75.00**	29.51	**45.90**	**52.46**	37.93	**62.07**	**65.52**
	$NoClose_1^{ST}$	35.00	56.67	**75.00**	**32.79**	**45.90**	**52.46**	**41.38**	55.17	**65.52**
	$NoStruc_1^{ST}$	20.00	48.33	56.67	29.51	**45.90**	**52.46**	34.48	44.83	48.28
	$EqWeights_1^{ST}$	30.00	48.33	66.67	27.87	**45.90**	**52.46**	34.48	55.17	62.07
Cross-lingual, S translated to one language	$AllInfo_1^{S}$	35.00	**56.67**	63.33	27.87	**42.62**	**47.54**	34.48	65.52	**68.97**
	$NoClose_1^{S}$	**38.33**	53.33	**66.67**	**29.51**	39.34	**44.26**	34.48	**68.97**	**68.97**
	$NoStruc_1^{S}$	28.33	53.33	53.33	**29.51**	40.98	**47.54**	**41.38**	51.72	55.17
	$EqWeights_1^{S}$	30.00	53.33	58.33	24.59	39.34	42.62	**41.38**	65.52	**68.97**
Cross-lingual, S translated to several languages	$AllInfo_n^{S}$	56.67	**78.33**	**86.67**	49.18	**70.49**	75.41	44.83	65.52	**72.41**
	$NoClose_n^{S}$	**58.33**	76.67	83.33	**54.10**	68.85	**77.05**	**48.28**	**68.97**	68.97
	$NoStruc_n^{S}$	46.67	70.00	81.67	44.26	68.85	75.41	**48.28**	58.62	65.52
	$EqWeights_n^{S}$	40.00	71.67	81.67	47.54	**70.49**	73.77	**48.28**	65.52	68.97
Cross-lingual, transfer	$AllInfo_n^{Str}$	45.67	76.67	85.00	39.34	67.21	72.13	41.38	62.07	**75.86**
	$NoClose_n^{Str}$	**53.33**	**80.00**	**88.33**	**47.54**	**70.49**	**75.41**	**51.72**	**72.41**	72.41
	$NoStruc_n^{Str}$	23.33	60.00	73.33	37.70	57.38	68.85	31.03	58.62	65.52
	$EqWeights_n^{S}$	40.00	71.67	81.67	**47.54**	**70.49**	73.77	**51.72**	65.52	65.52
	$AROMA$	38.33	–	–	4.92	–	–	3.45	–	–

Table 3. Results for monolingual, multilingual and cross-lingual matching scenarios

average score over all similarity features ($EqWeights$). In addition to this, we have aligned each pair of taxonomies with the state-of-the-art ontology aligner $AROMA$ [1], as it has been among the participants of the OAEI workshop series in the past years, and as it was available for download. In order to provide a level playing field for comparing the results, we have transformed all statements using custom label types to `rdfs:label` statements, and the hierarchical presentation information to `rdfs:subClassOf` (i.e. if `x` should appear above `y` in a financial statement, then `y rdfs:subClassOf x`) before applying $AROMA$.

4.5 Discussion

Impact of multilingual and cross-lingual labels. The results clearly indicate that the performance goes up for almost all matchers if multilingual information is available. This means that matching algorithms should be capable of leveraging information in all overlapping languages in S and T. Interestingly, this also seems to hold in cross-lingual scenarios, as the best results have been obtained in cross-lingual multilingual scenarios. This is further supported by the very high scores obtained in the cross-lingual transfer scenario.

Impact of structural features. In almost all scenarios, the setting which did not contain any structural information ($NoStruc$) performed considerably worse than the settings which used structural information. This is as such an expected result, as previous research has already shown the importance of structural information in ontology matching (see chapter 4.3 of Euzenat and Shvaiko [2]).

Impact of close matches. The settings ignoring close matches ($NoClose$) have performed consistently better than the $AllInfo$ setting in almost all scenarios. While this may seem counter-intuitive at first sight, there is a reasonable explanation for this. As close matches were also excluded from the test sets, the probability of assigning rank 1 to the exact match seems to be higher, as there are fewer candidates with high scores in the test set. As such, it seems reasonable to assume that the close matches occupy some of the higher ranks in the $AllInfo$ settings, which would need to be verified in future experiments. Moreover, manual analysis has revealed that close matches may even be less similar to a source concept than bad matches. For example, combinations of the ITCC source concept `TotaleAttivoCircolante` ("total current assets") with HGB target concepts show that the similarity scores for the bad match `bs.ass.fixAss` ("Fixed assets") are higher than those for the close match `bs.ass.other.comment` ("Other assets, disclosures"). While this may in principle be true for some combinations of concepts, it may as well be an undesired side-effect of the fact that the mapping work of the XEBR WG has not been completed yet, and that some of the bad matches are in fact close (or even exact) matches which have not yet been classified as such. On the one hand, this assumption can be verified by comparing the results of future experiments with the ones presented here. More importantly, however, the (supposedly) bad matches which have been ranked higher than exact or close matches can be used to speed up the work of the XEBR WG, by suggesting potential candidates not considered until now.

Comparison to baselines. Table 3 shows that the best setting in each taxonomy pair clearly outperforms $AROMA$. In the cases of XEBR/HGB and ITCC/HGB, $AROMA$ has performed extraordinarily low, which is surprising given the fact that the comparably reasonable score of 38.33% for XEBR/ITCC was obtained using exactly the same default configuration. A possible explanation for this is that the association rules approach followed by $AROMA$ does not work well for repetitive labels such as the ones found in FAS. The results of the naive baseline $EqWeights$ are surprising as well, though in the opposite respect. While the best setting outperforms the baseline in most scenarios, it is in some scenarios as good as the best setting or better. A possible explanation is that we have not attempted to optimise the learning parameters nor the training samples, and in fact, the baseline can be outperformed by adjusting the parameters.

Summing up these findings, it seems best to translate the labels in the source ontology to all languages available in the target ontology when trying to match a monolingual source ontology to a multilingual target ontology. Moreover, in most cases the settings in which both S and T have been translated perform better than the settings in which only S has been translated. This suggests that issues with translation quality as mentioned by Fu et al. [3] can to some extent be mitigated by translating to a pivot language. However, this claim still needs to be supported by further evidence from several language pairs, as the translation quality of MT systems varies greatly depending on the pair of languages considered (cf. [7]).

5 Related work

There have been a number of machine learning approaches to ontology matching, such as the ones by Ichise [4] and Nezhadi et al. [9]. In particular, Ichise also follows an SVM-based approach, and Nezhadi et al. evaluate different learned classifiers. In contrast to this, we approach the matching problem by assuming that good matches should be ranked higher than bad ones, instead of attempting to classify a specific pair of concepts as being either a match or no match.

Concerning multilingual and cross-lingual ontology matching, the SOCOM framework (Semantic-oriented cross-lingual ontology mapping; [3]) has presented an approach to cross-lingual ontology matching. Similar to what has been discussed in this paper, they first translate the source ontology to the language of the target ontology, and then apply monolingual matching strategies, which corresponds to cross-lingual ontology matching as defined in Definition 3a. above. However, we are not aware of any attempts to combine this with multilingual matching strategies (in the sense of Definitions 2 and 3c.), nor of an evaluation of different cross-lingual matching scenarios at a scale presented in this paper.

6 Conclusion and future work

In this paper, we have presented a novel approach to ontology matching that uses a ranking SVM to learn a matching function that ranks good matches between two ontologies higher than bad matches. In addition to this, we have provided a precise definition of multilingual and cross-lingual ontology matching in relation to monolingual matching, and tried to quantify their effects on the performance of different matching strategies. Our approach was evaluated on different pairs of financial accounting standards in different languages, simulating both monolingual, multilingual and cross-lingual scenarios. The results have shown that multilingual information can indeed improve the performance of ontology matching algorithms, even in cross-lingual scenarios.

As was mentioned above, further work should go into optimising the learning parameters of the ranking SVM, in order to arrive at an estimate for the optimal performance of the SVM-based approach. This optimsation could then be attempted for non-linear kernels as well, in order to be able to compare the results obtained with each kernel. In addition to this, we have tried to use a uniform sampling approach for all pairs of financial standards, although we have observed that the performance of some pairs of taxonomies can be improved when choosing different sampling strategies. As such, it seems reasonable to try to identify the characteristics of the set of training samples that produces optimal results for a given pair of ontologies, in order to improve the composition of the training set. Finally, we have so far neglected deeper linguistic information in the set of features. Here, it should be interesting to investigate the effects of including similarity measures which leverage e.g. the terminological or morphological structure of the labels.

Acknowledgements

This work has been supported by the European Union under Grant No. 248458 for the Monnet project. We thank Fabian Abel, Gilles Maguet, Susan Marie Thomas, Thomas Verdin, and the anonymous reviewers for their comments.

References

1. David, J., Guillet, F., Briand, H.: Matching directories and OWL ontologies with AROMA. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management. pp. 830–831. ACM, New York (2006)
2. Euzenat, J., Shvaiko, P.: Ontology matching. Springer (2007)
3. Fu, B., Brennan, R., O'Sullivan, D.: Cross-lingual ontology mapping - an investigation of the impact of machine translation. The Semantic Web, Lecture Notes in Computer Science 5926, 1–15 (2009)
4. Ichise, R.: Machine learning approach for ontology mapping using multiple concept similarity measures. In: Seventh IEEE/ACIS International Conference on Computer and Information Science, ICIS 08. Portland, Oregon (2008)
5. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (2002)
6. Jung, J.J., Håkansson, A., Hartung, R.: Indirect alignment between multilingual ontologies: A case study of Korean and Swedish ontologies. Agent and Multi-Agent Systems: Technologies and Applications, Lecture Notes in Artificial Intelligence 5559, 233–241 (2009)
7. Koehn, P.: Europarl: A Parallel Corpus for Statistical Machine Translation. In: Proceedings of the Tenth Machine Translation Summit. pp. 79–86. Phuket, Thailand (2005)
8. Levenshtein, V.I.: Binary Codes capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
9. Nezhadi, A.H., Shadgar, B., Osareh, A.: Ontology alignment using machine learning techniques. International Journal of Computer Science & Information Technology 3(2) (2011)
10. Trojahn, C., Quaresma, P., Vieira, R.: An API for multi-lingual ontology matching. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta (2010)
11. XBRL International Specification Working Group: Extensible Business Reporting Language 2.1. XBRL Recommendation (2008), http://www.xbrl.org/SpecRecommendations/