Light-Weight Cross-Lingual Ontology Matching with LYAM++

Abdel Nasser Tigrine, Zohra Bellahsene, Konstantin Todorov

Light-weight Cross-lingual Ontology Matching with LYAM++

Abdel Nasser Tigrine, Zohra Bellahsene, Konstantin Todorov

{lastname}@lirmm.fr
LIRMM / University of Montpellier, France

Abstract. During the last decade, several automatic ontology matching systems were developed to address the problem of ontology heterogeneity. Aligning cross-lingual ontologies is among the current challenging issues in the field of ontology matching. The majority of the existing approaches rely on machine translation to deal with this problem. However, inherent problems of machine translation are imprecision and ambiguity. In this paper, we propose a novel approach to the cross-lingual ontology matching task, relying on the large multilingual semantic network BabelNet as a source of background knowledge to assist the matching process. We have designed and tested a novel orchestration of the components of the matching workflow. Our approach is implemented under the form of a prototype named LYAM++ (Yet Another Matcher–Light)— a fully automatic cross-lingual ontology matching system that does not rely on machine translation. We report the results of our experiments that show that LYAM++ outperforms considerably the best techniques in the state-of-the-art according to the obtained results on the MultiFarm datasets of the Ontology Alignment Evaluation Initiative 2014.

Keywords: Ontology Matching, Background Knowledge, Cross-lingualisme

1 Introduction

Ontologies have become key elements in a variety of knowledge-based applications. However, they are continuously confronted with the problem of heterogeneity – syntactic, terminological, conceptual or semantic. Ontology matching techniques propose solutions to the heterogeneity problem by automatically discovering correspondences between the elements of two different ontologies and thus enabling interoperability [1,2].

In spite of the considerable progress that has been made in the field of ontology matching recently, many questions still remain open and many challenges to face — a complete overview can be found in [3]. The current work addresses the challenge of using explicit reference knowledge in order to make up for the missing background knowledge in the matching process. We apply this solution to a particular ontology matching problem — aligning cross-lingual ontologies, i.e., ontologies that are defined in different natural languages.

Indeed, considering multilingual and cross-lingual information is becoming more and more important, in view particularly of the growing number of content-creating non-English users and the clear demand of cross-language interoperability leading to the need of bringing multilingual semantic information and knowledge together in an explicit manner. In the context of the web of data, it is important to propose procedures for linking vocabularies across natural languages. Ontology matching techniques are also largely applied for data linking, or instance matching, where the problem of multilingualism appears even more often. Cross-lingual data and ontology matching is therefore a crucial task in order to foster the creation of a global information network, instead of a set of linguistically isolated data islands. However, as observed by Spohr *et al.* [4], most of the ontology alignment algorithms assume that the ontologies to be aligned are defined in a single natural language.

The methods that have been proposed to deal with cross-lingual ontology matching most commonly rely on automatic translation of labels to a single target language [5]. However, machine translation tolerates low precision levels and there is often a lack of exact one-to-one correspondence between the terms in different natural languages. Other approaches apply machine learning techniques [4] that usually require large training corpora that are rarely available in an ontology matching scenario.

We present LYAM++ (Yet Another Matcher - Light), a fully automatic cross-lingual ontology matching system making use of background knowledge to assist the matching process and to recreate the missing semantic context. Since we focus on the cross-lingual ontology matching problem, we rely on the multilingual semantic network BabelNet[1], which has the advantages of being openly available, large and general-purpose. Note that an alignment system is composed by a number of components, comprising usually a terminological matcher and a structural matcher, as well as certain filtering and verification modules [6]. The background knowledge provided by BabelNet is used in our approach within two of these components. In the first place, it is applied to evaluate the terminological similarities between the names of the ontological elements by reconstituting in a semantically coherent manner the label of a source entity in the language of the target ontology. In the second place, BabelNet is called upon within the structural matching component of the matching procedure. Note that the explicit background knowledge helps to reduce significantly the complexity of the similarity computation algorithms (wherefrom the word "light" in the name of our tool). Another original feature of our approach is the choice of orchestration of the components of the alignment workflow. Our experiments on the MultiFarm[2] benchmark data show that (1) our method outperforms the best cross-lingual matching approaches in the current state-of-the-art and (2) the novel workflow orchestration provides better results compared to the one that is commonly used by the established alignment systems.

[1] http://babelnet.org/
[2] http://web.informatik.uni-mannheim.de/multifarm/

In the following section, we focus on the technical aspects of our approach (Section 2). The experiments that have been conducted and their results are discussed afterwards (Section 3), followed by an overview of related approaches (Section 4) and a conclusion (Section 5).

2 Overview of the Approach

LYAM++ is a matching system specialized in dealing with cross-lingual ontology heterogeneity. LYAM++ makes use of several matching modules of YAM++ (an established and highly-performant tool [2], also developed by our research group), adding a reference knowledge component that aims to (1) provide the missing semantic context in the matching process and (2) lighten the alignment pipeline used in the original system. Particularly, LYAM++ makes use of the natural language processing module from YAM++ as well as several similarities measures. Indeed, one of our working hypotheses is that background knowledge, when used appropriately, can help to reduce the effort of matcher selection and tuning and can considerably shorten the chain of similarity measures that is commonly applied within the alignment processing workflow [6].

The workflow of LYAM++ is shown in Fig. 1. Let S and T be two input ontologies. Our goal is to align the former (*source*) to the latter (*target*). Additionally, we assume that S is given in a natural language l_S and T – in a language l_T. We have chosen BabelNet as a source of background knowledge and our processing pipeline uses two matchers: a multilingual terminological matcher (the main matcher), making use of only two similarity measures, and a structural matcher. In addition, we apply a mapping verification and selection filter.

2.1 A Novel Orchestration of the Workflow Components

In the following sections, we will describe each of the components mentioned above in details. Here, we draw the reader's attention to the first original contribution of our approach, which lies in the choice of orchestration of these components (Fig. 1). Note that most of the alignment tools perform terminological and structural matching before mapping selection and verification [6]. We reversed this order in an attempt to ensure that we "feed" only good quality mappings to the structural matcher. In that, LYAM++ filters the discovered correspondences right after producing the initial terminological alignment. The viability of this decision is supported experimentally in Section 3.

2.2 Preprocessing

As every ontology matching system, LYAM++ transforms the source and the target ontologies before applying the alignment procedure. The first preprocessing step performed by LYAM++ consists in splitting the elements of each ontology into three groups: labels of *classes*, labels of *object properties* and labels of *data object properties*, since these groups of elements are to be aligned separately.

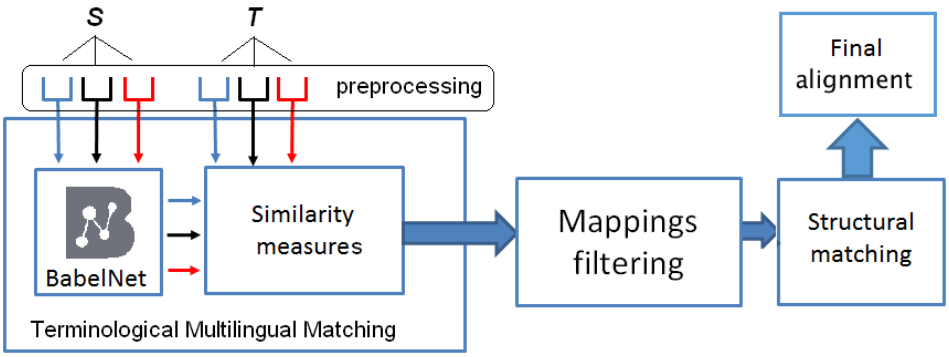


Fig. 1: The processing pipeline of LYAM++.

The labels of ontological elements are seen as strings of characters. In order to improve the result of comparing two strings, we apply a standard set of preprocessing procedures: normalization of characters and spaces, removing diacritics or accents, deleting numbers, punctuations and stop words, tokenization and lemmatization.

2.3 Background Knowledge: BabelNet

As stated above, our system makes use of multilingual background knowledge and our choice has fallen on BabelNet for several reasons. BabelNet is a multilingual semantic network and ontology that has been built by merging different encyclopedic and linguistic resources, such as the English and the multilingual WordNet, Wikipedia, the linked database Wikidata, and the multilingual dictionaries Wiktionary and OmegaWiki. The integration of these resources has been conducted automatically. BabelNet is openly available, large in scale and general-purpose resource, covering 271 languages. For these reasons, it appears to be an appropriate choice of background knowledge for cross-lingual ontology matching, although potentially our system is not restricted to this particular choice and could make use of any other multilingual reference knowledge base.

2.4 Terminological Similarity Measures

The main matching component of an alignment system (see following subsection) produces an intermediate alignment based on terminological similarities between the labels by calling on a string-based similarity measure. The ontology matching literature is rich in definitions of measures computing the degree of similarity between two labels. An exhaustive overview can be found in [7]. Our system makes use of only two similarity measures – a simple token similarity and a compound label similarity. The former is applied for measuring similarities between single-token strings while the latter is suited for labels composed of several tokens (like for example “Conference_Dinner”, “ConferenceDinner”,

"Conference-Dinner"). We will denote by s and t – two labels (strings of characters) to be compared, by s_i and t_j – tokens belonging to s and t, respectively, and by $|s|$ – the number of tokens in s.

Single-token Similarity Measures. An ontology matching system designer can choose among a large set of single-token similarity measures. Our choice has fallen on the Jaccard coefficient, which is based on the simple idea of estimating the relative overlap of two sets of objects. In our case, two tokens are seen as two strings of characters, s_i and t_j and the Jaccard coefficient is given as $sim(s_i, t_j) = \frac{|s_i \cap t_j|}{|s_i \cup t_j|}$, where the numerator represents the number of common symbols of the two strings and the denominator – the total number of symbols. Note that we have carried out experiments with edit-distance based measures, such as Levenshtein, but we retained the Jaccard coefficient because it showed to perform best in our experiments.

Similarity Measures for Compound Labels. This group of measures is based on first splitting the labels into the tokens that compose them, then applying a simple token similarity function (such as Levenshtein or Jaccard, denoted by *sim* in the following) internally before evaluating the overall label similarity. Our choice has fallen on three of the most common measures of this type.

(1) The **SoftTFIDF** measure is defined as a modification of the well-known cosine measure by using the TF/IDF weights [8]. We refer the reader to [7] for a formal definition. The TF/IDF weighting scheme is used to measure the relevance of a term with respect to a document and within a given corpus. In an ontology matching scenario, given a label composed by several tokens and an ontology composed of several labels, this scheme is applied by considering a token as a term, a label — as a document and the collection of labels in an ontology — as a corpus.

(2) The **Monge-Elkan** measure is defined as follows:

$$MongeElkan(s,t) = \frac{1}{|s|} \sum_{i=1}^{|s|} \max_{i=1}^{|t|} sim(s_i, t_j)$$

(3) The **Extended Jaccard** measure, as its name suggests, is based on the Jaccard similarity [7] given as follows:

$$ExtJaccard(s,t) = \frac{\mid C \mid}{\mid C \mid + \mid UniqueToken(s) \mid + \mid UniqueToken(t) \mid},$$

where $token(s)$ is a function returning the set of tokens composing a string s, $C = \{(s_i, t_j) | s_i \in token(s) \wedge t_j \in token(t) : sim(s_i, t_j) > \theta\}$ and $UniqueToken(s) = \{s_i | s_i \in token(s) \wedge t_j \in token(t) \wedge (s_i, t_j) \notin C\}$.

2.5 Main Multilingual Terminological Matching Using BabelNet

The main matcher is responsible for producing an initial *terminological* (i.e., based on the labels of the ontology elements) alignment that will be further on improved by the filtering modules and the structural matcher. This matcher uses in its algorithm the tools described in the previous subsections (BabelNet and two similarity measures). The multilingual matching is performed in two steps: (1) the labels of the source ontology are transformed into the language of the target ontology by using BabelNet and then (2) a monolingual label similarity computation is performed. In this process, two similarity measures are used – a simple similarity measure applied on single-token strings (in step 1) and a similarity function applied on compound labels composed by multiple tokens (in step 2). Our main contribution within this matching module lies in (1) a procedure for transforming the source labels into the language of the target ontology by using BebelNet and (2) a method for the semantic expansion of the transformed labels aiming to improve the final similarity values.

Transforming the Source Labels into the Target Language. In order to overpass the cross-lingual barrier, the tokens of the elements of T are transformed and enhanced by the help of BabelNet. At first, every token of a given label s in S is enriched by related terms and synonyms from BabelNet in the language l_T, which makes these terms comparable to the tokens of the labels in T. A simple similarity evaluation by the help of the Jaccard coefficient selects the term in each set of related terms corresponding to a given token from s that has the highest score with respect to every token in each label of T. This helps to re-constitute the label s in the language l_T. This procedure is presented in Algorithm 1. Finally, the labels in each group of S and T, seen as sets of tokens *in the same language (l_T)*, are compared to one another by using one of the measures described above (Soft TFIDF, Extended Jaccard or Monge-Elkan).

Example. We accompany the algorithm by an example given in Fig. 2. We look at the source label s={"chair of program committee"} given in English and the target label t={"président du comité de programme"}, given in French. After tokenization of s, the function *getSources*() is called for each of its tokens. This function queries BabelNet and returns a set of terms in the target language (French) related to the token it takes in its argument. In our example, we have

- *getSources*("chair", "EN", "FR")={Chair_acrobatics, maître_de_cérémonie, **président**, Fixation_des_rails_aux_traverses, professeur, fauteuil, Chaise_électrique, plomb, chaise}
- *getSources*("program", "EN", "FR")={Animatrix, logiciel, mission, plan, Programme_électoral, **programme**, programmer}
- *getSources*("committee","EN", "FR")={**comité**, Committee_(comics)}.

The resulting tokens are compared to the tokens of t pairwise by using the Jaccard measure, denoted by *sim* in the algorithm and in the figure. In order to

Algorithm 1 Transforming a label from its source language into a target language by using BabelNet.

Input: s, t: two labels
$s_i \in s, t_j \in t$: two tokens
l_S : language of s , l_T : language of t
$sim(s_i, t_j)$: a basic token similarity measure (e.g., Jaccard, Levensthein, or other)
Output: *BabelLabel*: the transformed label of s in the language l_T, as an outcome of BabelNet

```
1: for each s_i ∈ s do
2:    babelsource ← getSources(s_i, l_S, l_T)
3:    for each t_j ∈ t do
4:       for each b ∈ babelsource do
5:          score ← sim(b, t_j)
6:          if score > maxScore then
7:             maxScore ← score
8:             source ← b
9:          end if
10:      end for
11:      add(BabelLabel, source)
12:   end for
13: end for
```

constitute the French version of the source label s, denoted by $s_{\text{“}FR\text{”}}$, we replace each source token by the token returned by *getSources* that has scored best with respect to the target tokens (in bold in the example above). As a result, the transformed version of the source label takes the form $s_{\text{“}FR\text{”}} = \{$“président”, “programme”, “comité”$\}$. The final similarity score between s and t is an outcome of a compound label similarity function, like for example, the SoftTFIDF, computed for the labels $s_{\text{”}FR\text{”}}$ and t, that are now both in the same language (French, in our example).

Semantically Enhancing the Transformed Source Label Terms. In the case when the final similarity value produced by the softTFIDF measure for a given pair of labels is not satisfactory (i.e., is under a given threshold), Alg. 1 is called again by each of the tokens of the source label in its transformed form (an outcome of Alg.1), this time taking as an argument the token and two times the language l_T in order to enhance semantically the input information by looking for more related terms in BabelNet that might be better matching candidates. This process can be called upon as many times as desired, but in order to avoid complexity problems, we have limited the number of these iterations to two.

Example. Let us take a look at the source token $s_i =$ “*bid*”, and let $l_S =$ “EN” and $l_T =$ ”FR”, where “EN” stands for English and “FR” – for French. In this case, $getSources(s_i, l_S, l_T) = \{$Prixbid, souhaiter, offre, commande, offrir, Bid-TV, inviter, implorer, appeler$\}$. However, the right term for “bid” is “proposition” and it is not on the list. This results in flawed values of the final similarity

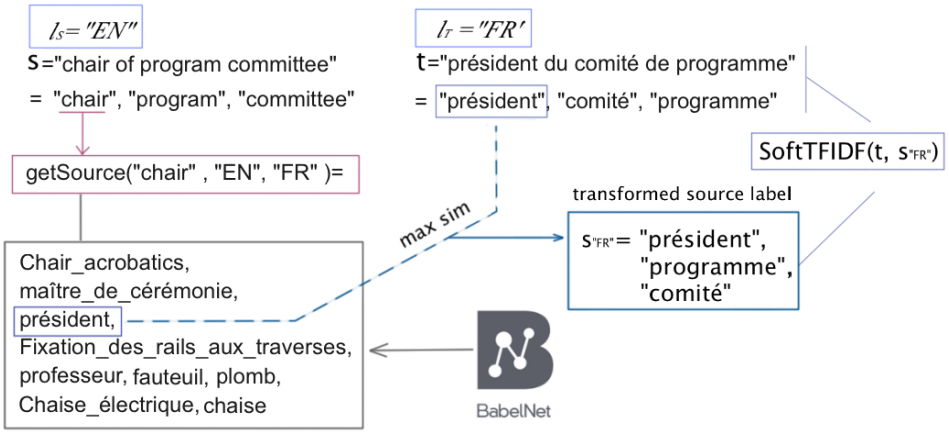


Fig. 2: Applying Alg. 1 – a transformation of a source label into the language of a target label by using BabelNet. An example with the token "chair".

between the source label, of which "bid" is part and any target label. To make up for that, since the terms in BabelNet are linked, we can map from the term "offre" to the term "proposition" by applying $getSources(\text{"offre"}, l_T, l_T)$.

Unlike most matching systems, which translate the labels of source and target ontology into English, LYAM++ transforms the labels of S from the language of S into the language of T. Note again that only one similarity measure is used on label level and one – on token level, which makes LYAM++ a light-weight ontology matching system.

2.6 Mapping Verification and Mapping Selection

The mapping verification step aims to remove correspondences that are less likely to be satisfiable based on the information present in the ontologies. This component filters out the trustworthy pairs of aligned concepts by looking at the similarity values produced for their parents and their children in the ontology hierarchies.

In an ontology matching system, mapping selection is an important task used as a filter to select high quality mapping candidates before producing the final alignment. This module transforms the initial 1 to many alignment (a source element possibly corresponding to multiple target elements) to a 1:1 alignment (a source element corresponds to exactly one target element) based on the principle of iteratively retaining the pairs of concepts with maximal value of similarity.

2.7 Structural Matching with BabelNet

Structural methods exploit the relations between entities, relying often on the hierarchical structure of the ontologies defined by the subsumption relation.

The basic idea is that if two entities are similar, their "relatives" could also be in one way or another similar. Two entities are considered similar if either of the following heuristics is true: (i) their direct super-entities (or all super-entities) are similar, (ii) all their sisters, who are the entities with the same super entity directly with the entities in question are similar, (iii) all of their sub-entities are similar, (iv) their descendants are similar, (v) all their leaves are similar.

In order to cope with the cross-lingual character of the input ontologies in the process of structural matching, we call upon BabelNet again. Note that the structural information is language independent, but a similarity measure is needed to verify the heuristics given above, and this similarity measure is most commonly language dependent, since it is based on similarity of strings. Our cross-lingual structural matching procedure is presented in Algorithm 2. Similarly to Algorithm 1, we query BabelNet to construct language transformations of the labels of the source ontology. Before computing similarity between the transformed source labels and the labels of the target ontology we check their structural information using the $MatchExist()$ function. The function $GetStructure()$ returns all structural information of an entity such as its super-entities and sub-entities. $MatchExist()$ returns true if any structural informations of the two entities are similar. It reduces the number comparisons between entities and thus optimizes considerably the matching process.

Algorithm 2 Structural matching with BabelNet

Input: e_1, e_2: two entities
s: label of e_1, t: label of e_2
$s_i \in s, t_j \in t$: tow tokens
– l_S : language of s , l_T : language of t
Output: $BabelLabel$: outcome of BabelNet

```
1: for each s_i ∈ s do
2:   if MatchExist(IA, getStructure(e1), getStructure(e2)) then
3:     babelsource ← getSources(s_i, l_S, l_T)
4:     for each t_j ∈ t do
5:       for each b ∈ babelsource do
6:         score ← sim(b, t_j)
7:         if score > maxScore then
8:           maxScore ← score
9:           source ← b
10:        end if
11:      end for
12:      add(BabelLabel, source)
13:    end for
14:  end if
15: end for
```

3 Evaluation and Results

We have evaluated LYAM++ on data coming from the ontology alignment evaluation initiative (OAEI)[3] of year 2014 and particularly Multifarm—a benchmark designed for evaluating cross-lingual ontology matching systems. Multifarm data consist of a set of 7 ontologies originally coming from the *Conference* benchmark of OAEI, translated into 8 languages. Two evaluation tasks are defined: *task 1* consists in matching two different ontologies given in different languages, while *task 2* aims to align different language versions of one single ontology.

We used the Alignment API[4] in order to compute precision, recall, and F-measures. We use BabelNet 3.0 API to query online BabelNet.

We have performed experiments on both tasks by using several MultiFarm datasets (the ontologies *CMT*, *conference*, *confOf, iasted* and *sigkdd*). We have conducted three experiments. In the first one, we compare LYAM++ to Agreement Maker Light (AML) [5] on both tasks and all pairs of languages. The choice of AML is motivated by the fact that, according to the reports of the OAEI 2014, this system performs best on the Multifarm track. In the second experiment, we evaluate the standard orchestration of the ontology matching workflow and the novel orchestration proposed in this paper. Finally, the third experiment compares the resuts obtained by each of the three similarity measures introduced in the previous section (TF/IDF, Extended Jaccard and Monge-Elkan) and shows that the performance of LYAM++ does not depend on the choice of measure, but is due to the use of background knowledge and the novel orchestration of the matching workflow.

3.1 Comparing LYAM++ to AML

The mapping threshold determines the value of the similarity function, above with a pair of labels can be considered as a potential mapping. It can also be seen as a confidence value of the produced alignment – the higher the mapping threshold, the more conservative we are in filtering out mappings and therefore the more confident we are in the final result. The F-measure depends strongly on this value and therefore, commonly, ontology matching results are presented in the form of an F-measure curve as a function of the mapping threshold.

Within this experiment, the SoftTFIDF similarity measure has been used to produce the initial terminological similarities. The Jaccard coefficient has been applied for the source label reconstitution in the target language (the measure *sim* within Alg. 1).

Table 1 shows the average F-measures over all threshold values per language pair for tasks 1 and 2. As it can be seen, LYAM++ outperforms AML systematically for all pairs of languages on both tasks, even for more difficult to handle languages like Russian. The high average F-measure values over all threshold values provide evidence of the stability of LYAM++ in terms of confidence value.

[3] http://oaei.ontologymatching.org/
[4] http://alignapi.gforge.inria.fr/

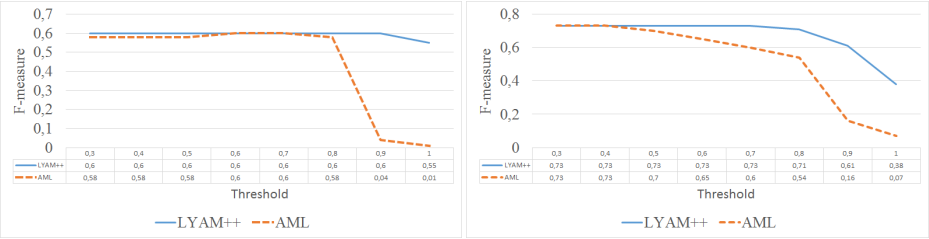
Table 1: Comparing LYAM++ to AML

Lang. pair	FR-RU	FR-PT	FR-NL	ES-FR	ES-RU	ES-PT	ES-NL	EN-PT	EN-RU	EN-FR
LYAM++	0.54	0.58	0.62	0.60	0.60	0.60	0.63	0.67	0.53	0.59
AML	0.48	0.51	0.47	0.53	0.47	0.51	0.52	0.53	0.51	0.49

Average F-measures over all threshold values per language pair for task 1.

Lang. pair	FR-RU	FR-PT	FR-NL	ES-FR	ES-RU	ES-PT	ES-NL	EN-PT	EN-RU	EN-FR
LYAM++	0.58	0.72	0.67	0.77	0.64	0.70	0.68	0.74	0.59	0.85
AML	0.44	0.64	0.57	0.66	0.51	0.66	0.61	0.68	0.48	0.70

Average F-measures over all threshold values per language pair for task 2.



(a) Task 1 (b) Task 2

Fig. 3: Average F-measures over all language pairs per threshold value and task.

Figure 3 shows the average F-measures over all language-pairs per threshold value and task. On task 1, both systems have similar and stable behavior for threshold values lower than 0.8. When this value is surpassed, however, one can see a clear advantage of LYAM++, which remains on more or less the same F-measure level, while the performance of AML drastically decreases reaching almost a 0 F-measure for a threshold values close to 1.

On task 2, the divergence in the performance of the two systems is observed at a much lower value—0.4—of the mapping threshold. For threshold values greater than 0.4, the values of the F-measure for AML start to decrease while LYAM++ remains stable until the threshold value reaches 0.8.

We explain the good results of our system by the appropriate use of a specific background knowledge source (BabelNet), particularly suited for cross-lingual ontology matching, as well as by the novel composition of the matching modules (see results of the experiment in the following subsection). We underline the fact that LYAM++ remains stable and outperforms AML for even high threshold values (close to or equaling 1), which demonstrates the capacity of the system to produce quality cross-lingual alignments with a very high level of confidence.

Table 2: Comparing the standard and the novel orchestrations

Langauge pair	FR-RU	FR-PT	FR-NL	ES-FR	ES-RU	ES-PT	ES-NL	EN-PT	EN-RU	EN-FR
Novel	0.58	0.72	0.67	0.77	0.64	0.70	0.68	0.74	0.59	0.85
Standard	0.50	0.58	0.60	0.39	0.54	0.57	0.58	0.50	0.32	0.39

Average F-measures over all threshold values per language pair for task 2.

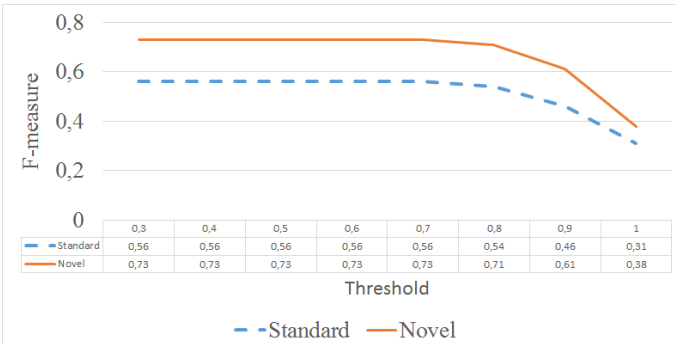


Fig. 4: Average F-measures over all language-pairs per threshold value of task 2 for the novel and the standard orchestrations.

3.2 Comparing the Standard and the Novel Orchestrations

This experiment aims to demonstrate the advantage of the novel orchestration of the components within the matching workflow as compared to the standard one. Again, the two similarity measures that are used are SoftTFIDF for the intermediate terminological alignment and Jaccard for the label reconstitution (Alg. 1). Following the same presentation pattern as the one in the subsection above, Table 2 shows the average F-measures over all threshold values per language pair, while Figure 4 shows the average F-measures over all language-pairs per threshold value. We have presented the results on task 2 only for reasons of space limitation. Similar results were obtained on task 1.

As we can see, the novel orchestration largely outperforms the classical one. This is due to the fact that the mappings resulting from the mapping selection module have high precision and these mappings are passed to the structural matcher, responsible for the final alignment.

3.3 Impact of the Choice of Similarity Measure

This experiment evaluates the impact of the choice of a similarity measure for comparing compound labels on the quality of the alignment produced with LYAM++. Fig.5 shows the average F-measures over all language-pairs per threshold value again for task 2. It can be seen that the F-measures obtained by the three evaluated similarity measures are not significantly different and

are all of good quality. This comes to show that our system is robust to the choice of similarity measure, its high performance being mainly due to the novel orchestration and the use of suitable background knowledge in the alignment process.

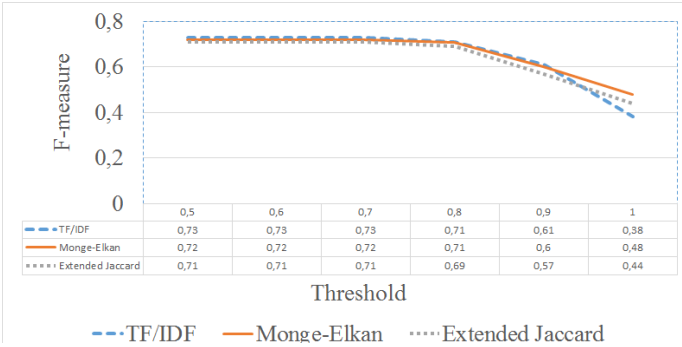


Fig. 5: Comparing the similarities measures.

4 Related work

The current section introduces related approaches to ontology matching with regard to two aspects: (1) the use of background knowledge and (2) handling cross-lingual heterogeneity.

4.1 Ontology Matching Using Background Knowledge

Using background knowledge (BK) for enhancing ontology matching is an idea that has been realized in several matching approaches in the literature. Sabou *et al.* [9] motivate the use of BK by the observation that two ontologies are always inherently different in terms of intention. Background knowledge comes to bridge the inherent semantic gap between them. We provide an overview of some of the relevant groups of techniques, categorized in terms of the type of background knowledge that is used, following the introduction found in [3].

Reusing existing mappings. Stored mappings can be used to discover new ones. Several approaches [10,11,12] follow this paradigm by reusing existing mappings to efficiently match two ontologies. The weak point of this group of approaches is the fact that they depend heavily on the quality of the re-used mappings, therefore on the performance of the ontology matching techniques that have been used to produce the initial mappings.

Using domain specific corpus. A corpus can be seen as a collection of elements (e.g., relation and attribute names) and their data types, relationships between elements, sample data instances, and other knowledge that can be used to discover mappings between entities [13,14]. Since a corpus is specific to a domain, it can only be used in specific matching cases. For example, a corpus in the field of anatomy can only be used for matching anatomy ontologies.

Using domain specific ontologies. Domain specific ontologies are often seen as quality sources of background knowledge. In [15,16], the alignment process takes place in two steps: *anchoring* and *driving relations*. Anchoring consists in matching the concepts of the source and target ontologies to the concepts of the reference knowledge using standard ontology matching techniques. In this step, each concept from the source and the target ontologies is mapped, or *anchored*, to a concept from the reference ontology. Driving relations is the process of finding relations between source and target concepts by checking if their corresponding anchored concepts are related.

Using the semantic web. This group of approaches is based on harvesting the semantic web in order to discover and explore multiple heterogeneous on-line knowledge sources [9]. The originality of the proposal is the use of the web in order to discover (by crawling) automatically appropriate BK sources (instead of using a single fixed source). Thus, the question of the availability and the coverage of the BK is addressed. This is particularly useful when the needed background knowledge is spread among different available sources.

Discussion. Due to the nature of the background knowledge that is used in our system, it is difficult to situate our approach in one of these families. Indeed, BabelNet can be seen as a corpus of existing mappings, since the terms are related semantically, but it can be also seen as an ontology. In contrast to the cited approaches, in our case the BK is used to enrich the semantic information contained in the source ontology in order to make it more "compatible" and easer to compare to the target ontology.

4.2 Cross-Lingual Ontology Matching

Gracia *et al.* [17] present a global *vision of a multilingual semantic web* together with several challenges to the multilingual semantic web community. According to the authors, multilingualism has to be seen as an extension of the semantic web– a group of techniques which will be added to the existing semantic technologies in order to resolve linguistic heterogeneity where it appears. The semantic web is seen as language-independent, because semantic information is given in formal languages. The main gap is, therefore, between language specific needs of users and the language-independent semantic content. The authors prognosticate that monolingual non-English linked data will increase in years creating "islands" of unconnected monolingual linked data. The challenge is to connect these islands by interconnecting the language-specific information. The authors outline the development of systems for establishing relations between

ontology terms or semantic data with labels and instances in different languages as a main direction of future research. We proceed to discuss different cross-lingual approaches grouped in four main categories depending on the underlying technique.

Machine translation (MT). The majority of approaches rely on MT techniques. Fu *et al.* [18] follow a standard paradigm of using monolingual matching techniques enhanced with an MT module. As a result of an analysis of the effect of the quality of the MT, the authors propose a noise-minimization method to reduce the flaw in the performance introduced by the translation. Trojahn *et al.* [19] have implemented an API for multilingual OM applying two strategies: a direct matching by a direct translation of one ontology to the other prior to the matching process and indirect matching, based on a composition of alignments. The latter approach is originally proposed by Jung *et al.* [20] and it is based on first establishing manual alignment between cross-lingual ontologies and then using these alignments in order to infer new ones. Paulheim *et al.* [21] apply web-search-based techniques for computing concept similarities by using MT for cross-lingual ontologies. Several well-established ontology matching systems propose to take cross-lingual ontologies as input by using machine translation in the alignment process, two of them being YAM++ [2] and AML [5].

Machine learning (ML). Spohr *et al.* [4] present an approach applying ML techniques. They use a small amount of manually produced cross-lingual alignments in order to learn a matching function for two cross-lingual ontologies. The paper introduces a clear distinction between a multilingual ontology (that which contains annotations given in different languages) and cross-lingual ontologies (two or more monolingual ontologies given in different natural languages).

Use of background knowledge. On the edge of the OM approaches that use background knowledge, Rinser *et al.* [22] propose a method for entity matching by using the info-boxes of Wikipedia. Entities given in different languages are aligned by the help of the explicit relations between Wikipedia pages in different languages. The matching relies mainly on the values of each property, since the actual labels are in different languages (e.g., "population" and "Einwohner" have approximately the same values (3,4M) in the info-boxes of the English and the German Wikipedia pages of Berlin). A very important and useful contribution of this paper is an analysis of the structure of the Wikipedia interlanguage links. Todorov *et al.* [23] [24] propose an ontology alignment framework based on background knowledge (the multilingual YAGO ontology) and fuzzy sets and logics to deal with imprecision in the cross-lingual matching process.

Natural language processing (NLP). Outside of the context of ontology matching, in the NLP field, research has been carried on the topic of measuring semantic distance between cross-lingual terms or concept labels. Mohammad *et al.* [25] and Eger *et al.* [26] propose measures of semantic distance between cross-lingual concept labels based on the use of bilingual lexicons. Explicit Semantic Analysis

(ESA) applied with Wikipedia has been proposed as a framework for measuring cross-lingual semantic relatedness of terms, first in a paper by Gabrilovich *et al.* [27] and then in an extended proposal by Hassan *et al.* [28]. It is suggested to rely on the multiple language versions of Wikipedia in order to measure semantic relatedness between terms. The authors use an ESA framework in order to model a concept as a vector in a space defined by a set of "encyclopedic concepts" in which the concept appears.

Discussion. The methods that have been proposed to deal with multilingualism in ontology matching, with few exceptions, rely on automatic translation of labels to a single target language. As noted in the introduction, MT tolerates low precision levels and often external sources are needed in order to achieve good performance. An inherent problem of translation as such is that there is often a lack of exact one-to-one correspondence between the terms across natural languages. Our approach is therefore closer in spirit to the approaches comping the NLP domain, the main idea being to use a bilingual reference vocabulary in order to link related terms across languages.

5 Conclusion

The moment to pay attention to cross-lingual ontology matching appears to be appropriate for several reasons. On the one hand, an important factor is the historical moment of the development of the Web community. As mentioned at the start, although originally predominantly English-speaking the Web, and consequently the Semantic Web, has the tendency of comprising more and more non-English active users, i.e., users that both consume *and* create Web content in languages other than English. In the current state of affairs there is only little above 30 percent of English Internet users[5] and the number of other language speaking users is constantly growing. In order to fully unlock the potential of the Web of Data project, the web community needs to be provided tools for the automatic integration of web knowledge—vocabularies and data— given in different natural languages.

In this paper, we have addressed this problem by proposing a novel cross-ontology matching approach implemented in the system LYAM++. In order to make up for the disadvantages of the currently existing cross-lingual matching systems, we do not rely on machine translation, but make use of background knowledge in the form of a large multilingual lexical network (BabelNet). In addition, we have proposed a novel orchestration of the matching components that form the ontology matching processing pipeline. We have shown experimentally by using data from the Multifarm benchmark, that our matching technique outperforms the best systems in the current state-of-the-art and that the novel workflow orchestration provides better results than the standard one.

In the future, we plan to explore the use of different kinds of background knowledge and the impact of this choice on the matching task. We will also

[5] http://www.internetworldstats.com/

apply our technique to the monolingual matching problem by exploiting the rich semantic information contained in a background knowledge source.

Acknowledgment

This work has been partially supported by the French National Research Agency (ANR) within the DOREMUS Project, under grant number ANR-14-CE24-0020 and the Ministry of Higher Education and Research of Algeria.

References

1. J. Euzenat and P. Shvaiko, *Ontology matching*. Heidelberg (DE): Springer-Verlag, 2007.
2. D. Ngo and Z. Bellahsene, "YAM++ : A multi-strategy based approach for ontology matching task," in *Knowledge Engineering and Knowledge Management - 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings*, pp. 421–425, 2012.
3. P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 1, pp. 158–176, 2013.
4. D. Spohr, L. Hollink, and P. Cimiano, "A machine learning approach to multilingual and cross-lingual ontology matching," in *The Semantic Web–ISWC 2011*, pp. 665–680, Springer, 2011.
5. D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto, "The agreementmakerlight ontology matching system," in *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, pp. 527–541, Springer, 2013.
6. D. Ngo, Z. Bellahsene, and K. Todorov, "Opening the black box of ontology matching," in *The Semantic Web: Semantics and Big Data*, pp. 16–30, Springer, 2013.
7. D. Ngo, Z. Bellahsene, and K. Todorov, "Extended tversky similarity for resolving terminological heterogeneities across ontologies," in *On the Move to Meaningful Internet Systems: OTM 2013 Conferences - Confederated International Conferences: CoopIS, DOA-Trusted Cloud, and ODBASE 2013, Graz, Austria, September 9-13, 2013. Proceedings*, pp. 711–718, 2013.
8. W. W. Cohen, P. D. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks," in *IIWeb*, pp. 73–78, 2003.
9. M. Sabou, M. d'Aquin, and E. Motta, "Exploring the semantic web as background knowledge for ontology matching," *J. Data Semantics*, vol. 11, pp. 156–190, 2008.
10. H. H. Do and E. Rahm, "COMA - A system for flexible combination of schema matching approaches," in *VLDB 2002, Proceedings of 28th International Conference on Very Large Data Bases, August 20-23, 2002, Hong Kong, China*, pp. 610–621, 2002.
11. A. Groß, M. Hartung, T. Kirsten, and E. Rahm, "GOMMA results for OAEI 2012," in *Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012*, 2012.
12. B. Saha, I. Stanoi, and K. L. Clarkson, "Schema covering: a step towards enabling reuse in information integration," in *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA*, pp. 285–296, 2010.

13. J. Madhavan, P. A. Bernstein, K. Chen, A. Y. Halevy, and P. Shenoy, "Corpus-based schema matching," in *Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03), August 9-10, 2003, Acapulco, Mexico*, pp. 59–63, 2003.
14. J. Madhavan, P. A. Bernstein, A. Doan, and A. Y. Halevy, "Corpus-based schema matching," in *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, pp. 57–68, 2005.
15. Z. Aleksovski, M. C. A. Klein, W. ten Kate, and F. van Harmelen, "Matching unstructured vocabularies using a background ontology," in *Managing Knowledge in a World of Networks, 15th International Conference, EKAW 2006, Podebrady, Czech Republic, October 2-6, 2006, Proceedings*, pp. 182–197, 2006.
16. Z. Aleksovski, W. ten Kate, and F. van Harmelen, "Exploiting the structure of background knowledge used in ontology matching," in *Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5, 2006*, 2006.
17. J. Gracia, E. Montiel-Ponsoda, P. Cimiano, A. Gomez-Perez, P. Buitelaar, and J. McCrae, "Challenges to the multilingual web of data," *Web Semantics: Science, Services and Agents on the World Wide Web*, 2011.
18. B. Fu, R. Brennan, and D. OŚullivan, "Cross-lingual ontology mapping–an investigation of the impact of machine translation," in *The Semantic Web*, pp. 1–15, Springer, 2009.
19. C. T. dos Santos, P. Quaresma, and R. Vieira, "An api for multi-lingual ontology matching," in *Procs of LREC*, 2010.
20. J. J. Jung, A. Håkansson, and R. Hartung, "Indirect alignment between multilingual ontologies: A case study of korean and swedish ontologies," in *Agent and Multi-Agent Systems: Technologies and Applications*, pp. 233–241, Springer, 2009.
21. H. Paulheim and S. Hertling, "Wesee-match results for oaei 2013," *Ontology Matching*, p. 197, 2013.
22. D. Rinser, D. Lange, and F. Naumann, "Cross-lingual entity matching and infobox alignment in wikipedia," *Inf. Syst.*, vol. 38, no. 6, pp. 887–907, 2013.
23. K. Todorov, C. Hudelot, and P. Geibel, "Fuzzy and cross-lingual ontology matching mediated by background knowledge," in *Uncertainty Reasoning for the Semantic Web III - ISWC International Workshops, URSW 2011-2013, Revised Selected Papers*, pp. 142–162, 2014.
24. K. Todorov, C. Hudelot, A. Popescu, and P. Geibel, "Fuzzy ontology alignment using background knowledge," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 22, no. 1, pp. 75–112, 2014.
25. S. Mohammad, I. Gurevych, G. Hirst, and T. Zesch, "Cross-lingual distributional profiles of concepts for measuring semantic distance.," in *EMNLP-CoNLL*, pp. 571–580, 2007.
26. S. Eger and I. Sejane, "Computing semantic similarity from bilingual dictionaries," in *Procs of the 10th International Conference on the Statistical Analysis of Textual Data (JADT-2010)*, pp. 1217–1225, 2010.
27. E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based esa.," in *IJCAI*, vol. 7, pp. 1606–1611, 2007.
28. S. Hassan and R. Mihalcea, "Cross-lingual semantic relatedness using encyclopedic knowledge," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pp. 1192–1201, Association for Computational Linguistics, 2009.