Ontolex JeuxDeMots and Its Alignment to the Linguistic Linked Open Data Cloud

Andon Tchechmedjiev, Théophile Mandon, Mathieu Lafourcade, Anne Laurent, Konstantin Todorov

Ontolex JeuxDeMots and Its Alignment to the Linguistic Linked Open Data Cloud

Andon Tchechmedjiev, Théophile Mandon, Mathieu Lafourcade, Anne Laurent, and Konstantin Todorov

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

Abstract. JeuxDeMots (JdM) is a rich collaborative lexical network in French, built on a crowdsourcing principle as a game with a purpose, represented in an ad-hoc tabular format. In the interest of reuse and interoperability, we propose a conversion algorithm for JdM following the Ontolex model, along with a word sense alignment algorithm, called JdMBabelizer, that anchors JdM sense-refinements to synsets in the *lemon* edition of BabelNet and thus to the Linguistic Linked Open Data cloud. Our alignment algorithm exploits the richness of JdM in terms of weighted semantic-lexical relations—particularly the inhibition relation between senses—that are specific to JdM. We produce a reference alignment dataset for JdM and BabelNet that we use to evaluate the quality of our algorithm and that we make available to the community. The obtained results are comparable to those of state of the art approaches.

Keywords: LLOD, Lexical resources, Lexical data linking, Ontolex, JdM

1 Introduction

The availability of large lexical-semantic resources (LSRs) is of central importance for a variety of natural language processing and semantic web applications. The lack of interoperability between these resources, as well as their limited coverage—most world languages are under-resourced to date—have been a significant hindrance to progress in the field.

JeuxDeMots (JdM) [1] is a collaborative lexical network of French terms built on a crowdsourcing principle as a game with a purpose (GWAP). JdM is very successful and has currently produced a network of over 1 million terms, more than 75 million relations of around 100 types, and around $70,000$ word senses for polysemous entries. Beyond its importance for French, JdM is a generic platform that can be adapted to other languages that critically require the production of LSRs. It is, therefore, an effective answer to the knowledge acquisition bottleneck.

However, JdM uses an ad-hoc tabulated data format, with a custom representation formalism that is different from typical (lexical architecture as opposed

to cognitive architecture) LSRs. Therefore, using JdM in conjunction with other resources is non-trivial and both JdM and its applications would benefit from being made interoperable.

With the advent of semantic web technologies, the Linguistic Linked Open Data (LLOD) [2], based on the *lemon* and Ontolex ontologies, is becoming a *de facto* standard for the access, interoperability and interlinking of language resources. Major state of the art LSRs such as BabelNet [3], Uby [4], many WordNets and DBnary [5] now exist as *lemon*/Ontolex[1] together with numerous alignments to other LSR datasets from the LLOD cloud.[2]

In light of the above, we address the problem of converting the JdM model to Ontolex and aligning it to the LLOD cloud. We use the core Ontolex model to represent entries and sense refinements (word senses), and the vartrans module of Ontolex to represent lexical and sense relations, to which we add a custom weight property. Given that JdM senses do not possess definitions (only word associations), linking JdM to another resource from the LLOD that is rich in definitions, would allow us to project the definitions back to JdM so as to enrich the network. We chose BabelNet as a target for the alignment as there already exist alignments between JdM and BabelNet at the lexical entry level. Additionally, BabelNet is one of largest resources on the LLOD cloud, possessing rich sense definitions. Given the structures of JdM and BabelNet, we developed a Word Sense Alignment (WSA) algorithm that we called JdMBabelizer, using a threshold decision criterion based on a weighted Lesk overlap similarity, where the weights of JdM relations and the normalized relative word frequencies of BabelNet definitions are taken into account. The proposed method is generic and language agnostic. Beyond its application to the data of the French lexical network, it can be seamlessly applied to editions of JdM in any other language. Thus, we enable the production of LLOD resources for languages such as Khmer, Thaï, Bangali, and Comorian, for which the JdM GWAP model has already been used.[3]

For the purpose of evaluating the JdMBabelizer algorithm, we construct a custom reference dataset by adding an innovate feature: we propose a crowdsourced gamified dataset creation, which considerably lowers the annotation burden. We make this benchmark dataset available to the community.

In the remainder of the paper, we first present JdM in detail followed by a related work review pertaining to the conversion of LSRs to *lemon*/Ontolex and to WSA techniques in the context of linking resources in the LLOD. Subsequently, we present the extended JdM/Ontolex model and the conversion algorithm, followed by a presentation of the WSA techniques applied. Before concluding, we evaluate the alignment with the help of our benchmark dataset.

[1] An exhaustive list of *lemon* resources: `https://datahub.io/dataset?tags=lemon`
[2] `http://linguistic-lod.org/llod-cloud`
[3] `http://jeuxdemots.liglab.fr`

2 JeuxDeMots: a Lexical Network

JeuxDeMots[4] (*Eng., word plays*) is a large lexical-semantic network, composed of terms (nodes) and typed, weighted and possibly annotated relations [1]. It contains term refinements (acceptations or word-senses), organized hierarchically (senses can have sub-senses). By May 15, 2017, it consists of roughly $75,799,080$ relations between $1,058,530$ nodes. Around $26,000$ polysemous terms are refined into $71,276$ word senses (or related usages for some domains). More than $800,000$ relations have negative weights and can be used as inhibitory relations. JdM contains links to DBnary and BabelNet at the word-level (words with the same canonical form). However, few alignments exist at the sense-level, although a dedicated tool allows the JdM players to refine word-level alignments.

2.1 Construction of JdM

JdM is a two player GWAP, allowing to *earn* and *collect* words. It has the following driving mechanics. (**1.**) The system (S) or a challenger (C) picks a term (T) that is offered to a player (A) along with a particular relation (R) from a manually curated list of relations (synonymy, antonymy, etc.) The system only chooses from existing terms, while challengers can offer new ones. (**2.**) Player A has a limited amount of time to enter terms which, to her/his mind, are related to T via R. The term T, along with the same set of instructions, will be later given to another player, say B, with an identical protocol for consolidation. The two players score points for words they both choose. The more "original" the proposition given by both players, the higher the reward. (**3.**) For a term offered to the players, answers in common from both A and B are inserted to the database (if the contributed terms are new, the term and a new relation instance are created with a weight of 1, otherwise the existing weights are incremented). Answers given by only one of the players are not considered, which reduces noise.

The network is constructed by connecting terms by typed and weighted relations, validated by pairs of players. Several other games complement the main JdM game.[5] Their purpose is to cross validate information collected in the main game, or to accelerate the relation harvesting for specific relation types.

2.2 Relations

An instance of each JdM relation links two particular nodes and has an associated weight. Relations can link nodes of any type. Even word-senses are defined as regular nodes that are linked to their corresponding entry by a particular type of refinement relation. Some lexical functions such as Magn and antiMagn[6] are represented as associative relations as well as predicative relations and can be

[4] `http://www.jeuxdemots.org`

[5] http://imaginat.name/JDM/Page_Liens_JDMv1.html

[6] Magn. for *Magnification* and antiMagn. the inverse relation: e.g. Magn(big)=huge, Magn(smoker)=heavy smoker, antiMagn(big)=small

in a sense equated to semantic frames. Although they represent the same type of information, they are encoded following the principles of the Meaning Text Theory (MTT) by Mel'čuk [6], rather than the semantic frame formalism (a conversion is non-trivial). The relations are not bound to grammatical categories (part of speech tags): grammatical categories are represented as separate nodes and linked to term (lexeme) nodes. The relations of JdM fall into one of the following categories.

– *Lexical relations:* synonymy, antonymy, expression, lexical family. This type of relations is about vocabulary and lexicalization.

– *Ontological relations:* hyperonymy, hyponymy, meronymy, holonymy, matter/substance, instances (named entities), typical location, characteristics and relevant properties, etc. These relations concern knowledge about world objects.

– *Associative relations:* free associations of feelings, meanings, similar objects, intensity (Magn and antiMagn). These relations are rather about subjective and global knowledge; some of them can be considered as phrasal associations.

– *Predicative relations:* typical agent, patient, instrument, or location of the action, typical manner, cause, or consequence. These relations are associated to a verb (or action noun) and to the values of its arguments (in a very wide sense).

Refinements. Word senses (or acceptations) of a given term node T (equivalent to a lexical entry) are represented as T>gloss$_1$, T>gloss$_2$, ..., T>gloss$_n$ nodes linked to the term node through REFINE(ment) relations. Glosses (following the lexicographical definition of gloss) are textual *annotations* that evoke the meaning of term T. For example, consider the French term *frégate (Eng., frigate)*. A frigate can be a ship or a bird (both English and French have the same ambiguity), and as a ship it can either be ancient (with sails) or modern (with missiles) (cf. upper part of Fig. 1 for an exmaple). Word refinements are structured, which, contrarily to a flat set of word meanings, has advantages for lexical disambiguation. Monosemous words do not have refinements as the term itself represents its only sense and requires no clarification.

Free Associations. The most common relation in the network, accounting for over 26% of all relations, is the free association relation (ASSOC), which for a given node provides cognitively related terms (mental associations). We make use of this relation to align JdM to other resources, as the terms related to a refinement through ASSOC form a sort of synset of words that allow humans to discriminate that particular meaning of the word and can thus be used as a substitution for definitions when overlap-based similarity measures are applied.

Inhibitory Relations. An inhibitory relation discriminates a specific refinement R_E of a top-level term E (equivalent to a lexeme/lexical entry) from another term T. Such a relation models the fact that if the term T negatively related to the R_E sense of E, appears in the same context as E (e.g. same sentence), then R_E is probably not the right sense for E in this context (relations of this type are extremely useful for Word Sense Disambiguation). Generally speaking, any relation between the refinement of a term and another term with a negative weight is inhibitory proportionally to its weight. However, there is also an explicit INHIB relation type, which indicates that the presence of the related term T

formally implies (with absolute certainty) that E cannot be in its R_E sense in that particular context. INHIB relations are computed automatically through the application of the following rule: $\forall E\ \exists T, R_{E,1}, R_{E,2}$: REFINE$(E, R_{E,1})$ $\wedge$ REFINE$(E, R_{E,2})$ $\wedge$ ASSOC$(R_{E,1}, T)$ $\wedge \nexists$ ASSOC$(R_{E,2}, T)$ $\Rightarrow$ INHIB$(T, R_{E,2})$. If the entry term E has at least two refinements, $R_{E,1}$ and $R_{E,2}$, and if the first refinement is associated to a term T but not the second one, then T inhibits the second refinement.

3 Related Work

Since the very early years of the web data field, rich LSRs have been called upon to provide robust semantic similarity measures [7], to assist ontology matching and link discovery across highly heterogeneous and multilingual datasets [8], [9], or to facilitate automatic question answering on large RDF graphs [10]. A crucial requirement to enable these applicaitons is that these resources are interoperable. In this section, we focus on the conversion of LSRs to RDF Ontolex and their interlinking on the web of data.

3.1 The Ontolex Model

Ontolex has emerged as a standard for representing lexical data on the web. It builds around the core model of its predecessor *lemon*, introduced by McCrae, Aguado-de-Cea, Buitelaar, *et al.* [11] to represent LSRs and their alignments with ontologies (OWL) and terminologies (SKOS), inspired by the LMF ISO-24613:2008 standard [12]. Ontolex adds modules for the representation of various linguistic phenomena and features (Syntax and Semantics, Decomposition, Variation and Translation, Linguistic Metadata, Linguistic Description, Lexical Networks).

For the representation of the JdM data, we are concerned with the use of the core model together with the Variational Translation (vartrans) module.[7] The main classes of the Ontolex core model include `LexicalEntry` and `LexicalSense`, the former representing the entry point into the resource (lemmatized words, affixes or multi-word expressions) and the latter representing word senses or semantic refinements associated to lexical entries. The `LexicalConcept` class allows to represent concepts lexicalised by a set of lexical senses and is a subclass of `skos:Concept`. The synsets in cognitive architecture LSRs (WordNet and derivatives, including BabelNet) would typically be represented by lexical concepts in Ontolex. The core model does not include the notion of lexical-semantic relations and we have to turn to the vartrans module to represent relations from resources such as BabelNet or JdM, through the reified `SenseRelation` class. Although `SenseRelation` does not have a weight data property, it is trivial to add one for the purpose of modeling the weights in JDM, for example.

Ontolex uses the `Form` class to describe the forms and representations of a `LexicalEntry`. Each lexical entry should have a canonical form, which is the

[7] `https://www.w3.org/community/ontolex/wiki/Final_Model_Specification`

lemmatisation of the term, and possibly other forms if any exist (e.g. morphological variants). Each form has a written representation datatype property that contains the terms. The linguistic meta-data module of Ontolex allows to encode useful information pertaining to lexical datasets, such as the language of lexical elements. The decomposition of multi-word expressions with relation to atomic lexical entries can be represented using the Decomposition module.

3.2 Converting Lexical Resources to Ontolex

Multiple LSRs built by professional linguists from scratch or by extending already exiting web resources have been successfully represented using lemon and its successor Ontolex, including Panlex [13], Parole [14], UBY [15], Eurosentiment [16] and Framebase [17]. In what follows, we focus on the main LSRs used in the web data field.

The well-known lexical database WordNet is composed of groups of quasi-synonyms called *synsets* with lexical relations linking synsets or words together. However, since the *lemon* model does not allow to represent synsets, in the *lemon* version of WordNet they have been represented as subclasses of `skos:Concept` linked to senses with the `lemon:reference` property [18]. Relations have been represented in the same way as in *lemon*UBY. Note that Ontolex now offers lexical concepts to represent synsets, while the vartrans module allows to describe relations directly (without using external vocabularies).

BabelNet [3], another well-established multilingual LSR, combines WordNet with Wikipedia (exploiting the multilingual information) and other resources (OmegaWiki, OpenMultilingualWordNet, etc.). Definitions in all languages are enriched through a machine translation of English definitions. The conversion of BabelNet to *lemon* [19] follows the same principle as that of WordNet, using the *lemon* vocabulary where possible along with other ontologies (*lime*, *lexvo*). The only custom class that had to be created in the conversion process is `BabelGloss`, representing the glosses bound to synsets.

Note that, unlike WordNet or BabelNet, JdM is created in a collaborative manner. Therefore, we pay close attention to DBnary [5], a LSR first modeled in lemon with custom properties, as it is also based on a collaborative resource (Wiktionary). We adopt a similar approach in the conversion of JdM to Ontolex and its alignment to the LLOD.

3.3 Word Sense Alignment Techniques for the LLOD Cloud

Although the LLOD cloud contains datasets represented as RDF graphs using the Ontolex ontology, aligning these resources is a substantially different problem as compared to standard data linking tasks on the larger LOD cloud. The problem we face here is that of aligning LSRs at the word sense level, known as Word Sense Alignment. Most linked resources in the LLOD cloud are aligned using techniques that are not specific to the LOD representation of the data, but to the pair of resources being aligned: there are no LOD specific algorithms for WSA.

WSA techniques use similarity between senses as a proxy for semantic equivalence across resources. The decision of whether to align two senses usually depends on an empirically determined threshold [20]. There are three main types of similarity computation approaches: lexical, structural, and hybrid. The field being vast, we only give several recent examples of applications relevant to this work. We refer the interested reader to [21].

Lexical similarity techniques exploit textual descriptions of lexical semantic elements (e.g. glosses or definitions) in LSRs. This is the most popular approach to WSA, as there are often definitions or some form of textual descriptions of senses in traditional LSRs (dictionaries). In recent applications, lexical similarity techniques have been applied (non exhaustively) to align the following resources (we provide the measures used and their performances in terms of precision (P), recall (R), F-score (F1) and accuracy(A) in brackets): Wiktionary and OmegaWiki (Personalized Page Rank (PPR) + Cosine (Cos) similarity, P 0.68, R 0.65, F1 0.66, A 0.78) [22]; WordNet and Wiktionary (PPR + Cos, P 0.67, R 0.64, F 0.66, A 0.91) [20]; GermaNet with Wiktionary (Lesk overlap measure, F1 0.84, A 0.91) [23]. Among the above-mentioned alignments, the most relevant to the present work is that of [23], as the authors apply an overlap-based measure using definitions. Moreover, one of their goals is to provide definitions to GermaNet from Wiktionary based on a projection through the alignments. Although the resources do not directly use lexical-semantic relations, these relations are present on Wiktionary pages and used to obtain extended textual representations for Wiktionary senses.

Structural similarity approaches exploit the topography of the graphs of the resources to determine whether two items should be aligned, by using classical graph search approaches and combinatorial graph metrics (path length, degrees, cycles, etc.). SSI-Dijkstra+ [24] has been applied to align WordNet and FrameNet (P 0.79, R 0.74, F1 0.75), while Dijkstra-WSA [22] — to align WordNet with Wikipedia (P 0.75, R 0.87, F1 0.81, A 0.95), as well as WordNet with Wiktionary (P 0.68, R 0.71, F1 0.69, A 0.92).

From a more general point of view, lexical and structural approaches can be combined in a *hybrid similarity framework*, by producing semantic signatures that include both definition-based and structural semantic information, normalized to live in the same space. This is the approach used to build resources such as BabelNet [3] or OpenMultilingualWordnet [25], formalized in an unified manner by Pilehvar and Navigli [26]. The framework remains the same for any resource pair and only the construction of the semantic signatures differs. Our extended overlap measure also enters in this category, as we create weighted bags of words (signatures) that contain words from definitions in BabelNet, related terms in JdM and the weights on relations from both resources.

The evaluation of the alignment of resources is tricky, because the reference data must be specific to each pair. Additionally, parameters that work for one pair, rarely generalize well to others. The standard approach in the domain is to either use an existing dataset to realign resources that are already aligned, or manually produce an evaluation dataset from a sample of representative entries.

We follow the latter approach, producing benchmark data in a novel crowdsourced game-based manner.

4 Producing Ontolex JeuxDeMots

Let us describe the conversion of the JdM tabulated (relational) model to Ontolex.

Core Model. The main elements in the core Ontolex model are lexical entries and lexical senses. We first identify corresponding elements in JdM. All nodes that are sources of a REFINE relation became lexical entries[8] and all nodes that are its targets became lexical senses.[9] We link corresponding lexical senses to their lexical entries and create `ontolex:Form` instances as needed to represent the canonical forms of the lexical entries.[10] A custom `jdm:id` datatype property contains the original JdM node id. Note that the hierarchical sense distinctions of JdM cannot be directly represented in Ontolex. We, therefore, do not represent sub-senses in the Ontolex model, only keeping the first level (cf. Fig. 1). For each lexical sense we create a lexical concept with a `lexinfo:gloss`

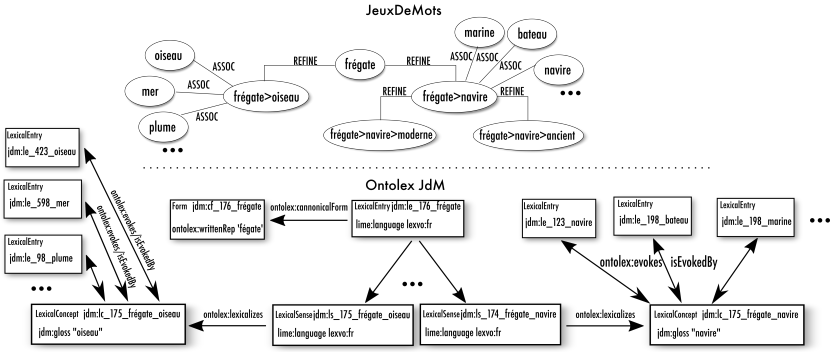

Fig. 1: An example of the conversion of term nodes and refinements to the Ontolex core model for the term *frigate*. Only first level senses are kept.

property that contains the gloss from the JdM refinement/sense node (it will be enriched with `skos:definition` in future work). For each ASSOC relation leaving from JdM refinement/sense nodes, the lexical concept that corresponds to that sense node is linked to the lexical entries of the corresponding words though the `ontolex:evokes/ontolex:isEvokedBy` property. In JdM, parts of speech are represented as POS nodes linked to terms. We retrieve the POS nodes for each lexical entry and add the `lexinfo:partOfSpeech` property.

[8] URI scheme *jdm:le_term*, where term is the canonical form of the term node.
[9] URI scheme *jdm:ls_term_gloss*, where gloss is the gloss of the refinement node.
[10] URI scheme *jdm:cf_term*, where term is the canonical form of the term node.

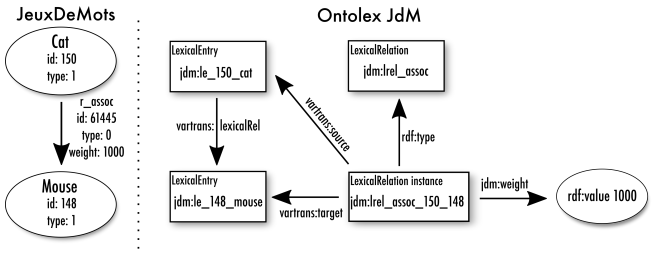

Fig. 2: Example of an association between *cat* and *mouse* in JdM (left) and its equivalent in Ontolex JdM (right).

Relations and Vartrans. What remains to be modeled are the numerous specific relations found in JdM, from which we exclude relations encoding structural information, used in the previous section. For that task, we turn to the vartrans module of Ontolex. Each relation is represented as a sublcass of `ontolex:LexicalRelation` and/or `ontolex:SenseRelation` as there are relations at both levels.[11] Where possible, we also made the relations sub-classes of existing relations in DBnary or in SKOS (OWL allows multiple inheritance). We added a custom `jdm:weight` datatype property to relation instances to represent `jdm:weights`.[12] The ASSOC relations are represented by `ontolex:evokes` and `ontolex:isEvokedBy` but also have weights, which cannot be represented by `ontolex:LexicalRelation` nor by `ontolex:SenseRelation`. We reify the `ontolex:evokes/isEvokedBy` as a sub-class of `ontolex:LexicoSemantic Relation` directly, as the source and targets can be `LexicalSense/LexicalEntry` or the reverse. We also represent weights by the `jdm:weight` relation. Fig. 2 illustrates how a lexical relation is represented in the original JdM data and in its Ontolex version. We make the converted JdM data available.[13]

5 Linking Ontolex JdM to the LLOD Cloud

We aim at producing alignments of JdM to other lexical or ontological resources published on the LLOD cloud at the level of lexical senses. JdM has no definitions, but the glosses provide some information as do the numerous ASSOC links that evoke the lexical senses. We can thus produce textual descriptions that capture the semantics of the lexical senses that can be used for WSA.

JdM already contains alignments at the lexical entry level to other LSRs (DBpedia, DBnary, BabelNet) and to certain medical ontologies (Radlex, UMLS) through ad-hoc approaches and in ad-hoc formats that are not interoperable with the LLOD cloud. We may thus reuse the alignments as a starting point to

[11] URI scheme: *jdm:lr_relname* or *jdm:sr_relname*.
[12] URI scheme of relation instances: *lri_sourcenodeid_targetnodeid* or *sri_sourcenodeid_targetnodeid*.
[13] `https://tinyurl.com/jdmbabelnetbench`

Algorithm 1 JdMBabelizer: the JdM/BabelNet alignment algorithm

```
function JdMBabelizer(jdmLE, bnLE, inhib={None/Vt/Wgt}, θ)
    alignedPairs ← ∅
    for all jdmS ∈ Senses(jdmLE) do
        for all bnS ∈ Senses(bnLE) do
            bnSig ← ∅
            AddToSig(bnSig, Words{Def(bnS)}, w = 1.5)
            AddToSig(bnSig, Lemma{Senses(Synset(bnS))}w = 2.0)
            AddToSig(bnSig, BibTaxonomy(bnS), updtW = 1.5)
            AddToSig(bnSig, Words{Examples{bnS}}, w = 0.75)
            jdmSig ← ∅
            for all evokedLe ∈ EvokedBy(jdmS) do
                AddToSig(jdmSig, {WrRep(evokeLe)},
                                 w = rWeight(jdmS, evokedLe))
            end for
                                              ▷ Weight-based inhibition strategy
            if inhib = Wgt then
                for all inhibLe ∈ Inhib(jdmS) do
                        ▷ Adding term to signature with the largest negative weight
                    AddToSig(jdmSig, {CanWrRep(inhibLe)}, w = −1000)
                end for
            end if
                    ▷ If there isn't an inhibition while in veto mode, we continue
                                          ▷ Otherwise, we veto this pair of senses
            if inhib ≠ Vt ∨ ∄t ∈ bnSig ∩ jdmSig : WrRep(Inhib(jdmS) = t) then
                score ← 0
                for all ∀bnSigEl, jdmSigEl ∈ words(jdmSig) ∪ words(bnSig) do
                    score ← score + weight(bnSigEl) × weight(jdmSigEl)
                end for
                if score > θ then
                    CreateAlignement(jdmS, bnS)
                end if
            end if
        end for
    end for
end function
```

align the resources at the lexical sense level through explicit RDF statements so as to include JdM in the LLOD cloud.

As a first step, we endeavour to align JdM to BabelNet, as BabelNet has rich definitions in several languages, that we could project back unto JdM through the alignment. We start off at entry level alignments and then compare all (BabelSense, JdM LexicalSense) pairs to find the ones that are most likely to be equivalent. Algorithm 1, named JdMBabelizer, details the process, roughly following the approach formulated by Pilehvar and Navigli [26].

For each of the pairs, we create a weighted bag-of-words semantic signature for the BabelNet sense and another for the JdM sense. For the BabelNet sense,

we build the signature from the words of the definition, the lemmas of the other senses corresponding to the synset of the sense, the category names from the the Wikipedia Bitaxonomy [27] and the words from the examples. We keep only unique words and increment the weight associated to each word (+1.5 if the word comes from a definition, +2.0 if the word comes from the lemmas of the synset senses, +1.5 for BibTaxonomy categories and + 0.75 for words from the examples) by using the `AddToSig` function that takes the existing signature, a set of words to add and an update weight (lines 6 to 9). `AddToSig` filters stop words and lemmatizes the words to add before their addition.

We create the signature for JdM by taking all the canonical written representations of lexical entries that evoke the sense (initially, the ASSOC relation), where the weights of each word correspond to the normalized relation weight (a value between -1000 and 1000). We reuse the same `AddToSig` function (10-14). In the case of the weight-based inhibition strategy, we add each word stemming from an inhibition relation to the signature with the highest negative weight (-1000, lines 15-20).

If we are in a veto inhibition mode and if there is an inhibition relation that points to a lexical entry that has a written representation matching words from the BabelNet signature, we immediately discard the current pair of senses (line 24). Otherwise, we move on to the score computation: for each overlapping word between the BabelNet signature and the JdM signature, we increment the score by the multiplication of the weight for the word from the BabelNet signature and the weight for the word from the JdM signature (lines 25-28). This is a weighted Lesk overlap similarity measure [28]. If the score is higher than the threshold, we create the alignment by adding a triple to the RDF model (lines 29-32).

6 Evaluation of the Linking Algorithm

The current section presents an evaluation of our linking approach. We start by describing our benchmark data before presenting and analyzing our results.

6.1 Benchmark Construction

Due to the specificity of JdM, it is difficult to use off-the-shelf benchmark data to evaluate our linking algorithm. Therefore, we manually create our own benchmark (as is customary in the field), containing valid links between JdM and BabelNet. To this end, we created a new game within a crowdsourcing paradigm. For two corresponding entries in JdM and BabelNet (same lemmas), the game shows to the player all of the BabelNet senses and for each of them a list of possible sense refinements from JdM (word senses). The player can click on each of the JdM refinements to mark the correspondence as true, false or undefined.[14]

[14] A link to the game with an example of the word "chat" (*Fr., cat*): `http://www.jeuxdemots.org/aki_fech_babelnet_distrib.php?term=chat`

Since JdM, contrarily to BabelNet, has case sensitive entries, it is useful to be able to say that a given synset does not match the JdM entry. For that purpose, all synsets containing, e.g., "jade" will be returned for both "Jade" (with one sense being the first name), and "jade" (one sense of this being the gem). Approximately half of the benchmark dataset contains inhibition relations. We prioritize words with many senses and many matching BabelNet synsets (common words like "cat"). Since there are approximatively 25,000 polysemous words in JdM, we included the hardest cases in order to have an overview of the worst-case alignment scenarios. We also picked nouns with few outgoing relations, like the French "religieuse", which can both be a religious person and a kind of a pastry, to analyse the impact of a lack of information on the the alignment results.The resulting dataset contains 574 links between nouns, accounting for approximately 2.5% of all possible links. It is used for all of our experiments and made freely available.[15]

6.2 Experimental Protocol

We start by selecting all noun nodes in JdM that are not refinements and that have at least two distinct semantic refinements (senses). Then, we compare and decide whether to align the semantic refinements of each of these terms to all of the BabelNet senses of nouns that have the same written representation[16], through the application of the JDMBabelizer algorithm. Subsequently, we evaluate the results against our benchmark data.

We ran Algorithm 1 for the entire JdM on a Hitachi HTS547575A9E384 laptop, with 8G RAM and an i5-2450m 2.50GHz processor. The final alignments, as well as both JdM and BabelNet Lucene indexes were stored on a mechanical hard-drive. There are 19782 polysemous nouns in JdM with a total of 51657 senses that we compare to 58816 tentative equivalent BabelSynsets. The entire alignment process took 4927597ms to run (approximately 1h21min). The solution space for the alignment is the union of Cartesian products of lexical senses for each pair of aligned lexical entries.

6.3 Results and Discussion

Threshold tuning. We start by estimating the optimal value of the cutoff threshold. We show the results for several threshold values in Table 1. Two scenarios stem from this experiment: (s1) favoring recall, with a corresponding threshold of 500 and (s2) favoring precision, with a threshold of 1,000. Although we give more importance to (s2) (ensuring that the established links are mostly correct), we analyze both cases in detail below with regard to the effects of inhibition.

Impact of inhibition. The results of our experiments on both scenarios by using inhibitions as negative weights, using inhibition as a veto (if an inhibited

[15] `https://tinyurl.com/jdmbabelnetbench`

[16] We used the BabelNet API `http://babelnet.org/guide`

Threshold	Precision	Recall	Accuracy
500	66%	80%	93%
750	68%	65%	93%
1,000	74%	51%	93%
1,250	74%	47%	91%

Table 1: Threshold variation.

word is found, the link is immediately discarded) and not using inhibition are shown in Table 2 in terms of Precision ($\frac{TP}{TP+FP}$), Recall ($\frac{TP}{TP+FN}$), F-measure (harmonic mean of P and R) and Accuracy ($\frac{TP+TN}{TP+FP+TN+FN}$). With a threshold of 500 (s1), we achieve an uninhibited Precision of 65% with a recall at 80% and a F-measure of 72%, which translates into an accuracy of 93%. With a threshold of 1,000 (s2), we achieve an uninhibited Precision of 73% with a recall at 52% and a F-measure of 60%, which translates into an accuracy of 92%. When we take inhibition into account as negative weights, we increase precision by 1%, while the other measures remain the same in both (s1) and (s2). When we take inhibition into account as a veto, we increase precision by 2% but decrease recall by 2% in (s2) and by 4% in (s1). For (s1), the F-measure decreases by 1% with no impact on accuracy, while in (s2) the F-measure remains unchanged, but the accuracy increases by 1%. All-in-all, the impact of inhibition appears to be much less significant than what we anticipated. However, in the interest of producing the most reliable alignment between JdM and BabelNet (at the price of lower recall), we identify the best configuration to be (s2) with a veto inhibition.

Dataset	Threshold/Scenario	Precision	Recall	F-measure	Accuracy
NoInhib	500 / (s1)	65%	80%	72%	93%
Inhib	500 / (s1)	66%	80%	72%	93%
InhibVeto	500 / (s1)	67%	76%	71%	93%
NoInhib	1000 / (s2)	73%	51%	60%	92%
Inhib	1000 / (s2)	74%	51%	60%	93%
InhibVeto	1000 / (s2)	76%	49%	60%	93%

Table 2: Results of aligning JdM to BabelNet with and without using the inhibition relation and by using it as a veto.

Error analysis. In order to better understand our results, we studied the false negatives and false positives produced by our algorithm. As expected, many false negatives are due to lack of information in JdM. For instance, one of the senses of the French word “baguette” is a rod used to push ammo for old firearms. The JdM entry contains only three outgoing relations, two of them being “military” and “history”, while the BabelNet synset does not mention neither of these. Since JdM is constantly evolving thanks to the permanent contributions of its players, we can hope that this missing information will be filled in the future.

The participative nature of JdM has also its downsides. Certain false negatives are due to the fact that the BabelNet synsets tend to contain academic definitions, while the terms linked through the JdM associations are rather common or colloquial.

Another source of false positives lies in the fact that some synsets do not have French definitions and use English ones instead. Since JdM is only in French, we want the projected definitions to be in French, too. For that reason, we systematically discard links to definitions in other languages. However some of these links are still established by our algorithm, because certain words have the same written representation in both languages. For example, the English definition of "devil" contains "cruel" and "demon", both valid French words and present in the JdM relations.

Among the remaining false positives, we frequently encounter senses that are close but still distinct. For example, "copper" can be used to describe the metal, or the color. Since the color is called that way because it is the color of the metal, these senses are tightly related and mislead the similarity judgment. This problem could be resolved by using more specialized relations in both BabelNet and JdM, like the *is_a* relation.

Comparison to state of the art. Comparing WSA results directly to the state of the art is generally difficult, because each time a specific pair of resources are aligned, having specific properties and evaluated on different reference datasets. This difficulty is amplified in our case by the lack of definitions in JdM. Nonetheless, we note that the best results obtained for scenarios (s1) and (s2), respectively, outperform the average of the WSA approaches. In scenario (s2), we obtain significantly higher precision values than most established approaches. The benefits of using the inhibition relation become clear as it adds a combinatorial pruning constraint that improves precision, although it decreases recall. In turn, this explains why the impact of inhibition is marginal in scenario (s1).

7 Conclusion

This paper deals with the addition of JdM, a French lexical resource, to the linguistic web of data. We introduce a conversion scheme of JdM to RDF allowing to model weighted relations by using Ontolex along with an approach to link JdM to BabelNet and thus to the LLOD. These links can be used for automatic translation, or to help enrich BabelNet using the JdM data and vice versa, enabling the interoperability of the two resources. By adding JdM to the LLOD, we also contribute to the enrichment of non-English linguistic resources on the web. We construct a benchmark dataset in the form of a reference alignment between JdM and BabelNet on the basis of a crowdsourced game. We use this data for evaluating our approach and we share it along with all produced data and algorithms.

References

[1] M. Lafourcade, "Making people play for lexical acquisition with the jeuxdemots prototype," in *SNLP'07: 7th international symposium on natural language processing*, 2007, p. 7.

[2] C. Chiarcos, S. Hellmann, and S. Nordhoff, "Towards a linguistic linked open data cloud: The open linguistics working group.," *TAL*, vol. 52, no. 3, pp. 245–275, 2011.

[3] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.

[4] J. Eckle-Kohler, J. P. McCrae, and C. Chiarcos, "Lemonuby a large, interlinked, syntactically-rich lexical resource for ontologies," *Semantic Web*, vol. 6, no. 4, pp. 371–378, 2015.

[5] G. Sérasset, "Dbnary: Wiktionary as a lemon-based multilingual lexical resource in rdf," *Semantic Web*, vol. 6, no. 4, pp. 355–361, 2015.

[6] I. Mel'čuk, "Lexical functions in lexicography and natural language processing," in, L. Wanner, Ed. J. Benjamins, 1996, ch. Lexical functions: a tool for the description of lexical relations in a lexicon, pp. 37–102.

[7] I. Hulpuş, N. Prangnawarat, and C. Hayes, "Path-based semantic relatedness on linked data and its use to word and entity disambiguation," in *International Semantic Web Conference*, Springer, 2015, pp. 442–457.

[8] P. Shvaiko and J. Euzenat, "Ontology matching: State of the art and future challenges," *IEEE Transactions on knowledge and data engineering*, vol. 25, no. 1, pp. 158–176, 2013.

[9] A. N. Tigrine, Z. Bellahsene, and K. Todorov, "Light-weight cross-lingual ontology matching with lyam++," in *ODBASE: Ontologies, DataBases, and Applications of Semantics*, 2015, pp. 527–544.

[10] C. Unger, A. Freitas, and P. Cimiano, "An introduction to question answering over linked data," in *Reasoning Web International Summer School*, Springer, 2014, pp. 100–140.

[11] J. McCrae, G. Aguado-de-Cea, P. Buitelaar, P. Cimiano, T. Declerck, A. Gómez-Pérez, J. Gracia, L. Hollink, E. Montiel-Ponsoda, D. Spohr, and T. Wunner, "Interchanging lexical resources on the semantic web," *Language Resources and Evaluation*, vol. 46, no. 4, pp. 701–719, Dec. 2012, ISSN: 1574-0218. DOI: 10.1007/s10579-012-9182-3. [Online]. Available: https://doi.org/10.1007/s10579-012-9182-3.

[12] G. Francopoulo, N. Bel, M. George, N. Calzolari, M. Monachini, M. Pet, and C. Soria, "Lexical Markup Framework (LMF) for NLP Multilingual Resources," in *Workshop on Multilingual Language Resources and Interoperability*, ACL, 2006, pp. 1–8.

[13] P. Westphal, C. Stadler, and J. Pool, "Countering language attrition with panlex and the web of data," *Semantic Web*, vol. 6, no. 4, pp. 347–353, 2015.

[14] M. Villegas and N. Bel, "Parole/simple lemon ontology and lexicons," *Semantic Web*, vol. 6, no. 4, pp. 363–369, 2015.

[15] I. Gurevych, J. Eckle-Kohler, S. Hartmann, M. Matuschek, C. M. Meyer, and C. Wirth, "Uby - a large-scale unified lexical-semantic resource based on lmf," in *13th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, 2012, pp. 580–590.

[16] P. Buitelaar, M. Arcan, C. A. Iglesias Fernandez, J. F. Sánchez Rada, and C. Strapparava, "Linguistic linked data for sentiment analysis," in *2nd Workshop on Linked Data in Linguistics: Representing and linking lexicons, terminologies and other language data*, Telecomunicacion, 2013.

[17] J. Rouces, G. de Melo, and K. Hose, "Framebase: Representing n-ary relations using semantic frames," in *European Semantic Web Conference*, Springer, 2015, pp. 505–521.

[18] J. McCrae, C. Fellbaum, and P. Cimiano, "Publishing and linking wordnet using lemon and rdf," in *3rd Workshop on Linked Data in Linguistics*, 2014.

[19] M. Ehrmann, F. Cecconi, D. Vannella, J. P. McCrae, P. Cimiano, and R. Navigli, "Representing multilingual data as linked data: The case of babelnet 2.0.," in *LREC*, 2014, pp. 401–408.

[20] C. M. Meyer and I. Gurevych, "What psycholinguists know about chemistry: Aligning wiktionary and wordnet for increased domain coverage," in *5th International Joint Conference on Natural Language Processing*, Asian Federation of Natural Language Processing, 2011, pp. 883–892.

[21] I. Gurevych, J. Eckle-Kohler, and M. Matuschek, *Linked lexical knowledge bases: Foundations and applications*. Morgan & Claypool, 2016.

[22] M. Matuschek, "Word sense alignment of lexical resources," PhD thesis, Technische Universitat Darmstadt, 2015.

[23] V. Henrich, E. Hinrichs, and T. Vodolazova, "Aligning germanet senses with wiktionary sense definitions," in *Human Language Technology Challenges for Computer Science and Linguistics: 5th Language and Technology Conference, Revised Selected Papers*. Springer, 2014, pp. 329–342.

[24] E. Laparra, G. Rigau, and M. Cuadros, "Exploring the integration of wordnet and framenet," in *5th Global WordNet Conference*, 2010.

[25] F. Bond and R. Foster, "Linking and extending an open multilingual wordnet," in *51st Annual Meeting of the Association for Computational Linguistics*, ACL, 2013, pp. 1352–1362.

[26] M. T. Pilehvar and R. Navigli, "A Robust Approach to Aligning Heterogeneous Lexical Resources," in *52nd Annual Meeting of the Association for Computational Linguistics*, ACL, 2014, pp. 468–478.

[27] T. Flati, D. Vannella, T. Pasini, and R. Navigli, "Multiwibi: The multilingual wikipedia bitaxonomy project," *Artificial Intelligence*, vol. 241, pp. 66–102, 2016.

[28] M. Lesk, "Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone," in *5th Annual International Conference on Systems Documentation*, 1986, pp. 24–26.