Harnessing Open Information Extraction for Entity Classification in a French Corpus

Fabrizio Gotti, Philippe Langlais

RALI, Université de Montréal, CP 6128 Succursale Centre-Ville, Montréal, Canada, H3C 3J7
{gottif,langlais}@iro.umontreal.ca

Abstract. We describe a recall-oriented open information extraction system designed to extract knowledge from French corpora. We put it to the test by showing that general domain information triples (extracted from French Wikipedia) can be used for deriving new knowledge from domain-specific documents unrelated to Wikipedia. Specifically, we can label entity instances extracted in one corpus with the entity types identified in the other, with little supervision. We believe that the present study is the first one that focusses on such a cross-domain, recall-oriented approach in open information extraction.

Keywords: natural language processing · open information extraction · named entities · entity classification

1 Introduction

Extracting knowledge from a large set of mostly unstructured documents (such as the Web) and organizing it into a knowledge base (KB) is a key challenge in artificial intelligence [1]. Intuitively, such KBs should directly impact the quality of many NLP applications such as question answering or information retrieval. Open information extraction (OIE), the task of extracting knowledge from texts without much supervision (especially not a prescription of the kind of information to mine), has brought new hope for such an endeavour. It has given rise to a number of exciting realizations, many fostered by major search engine companies. One of the most striking projects is IBM's Watson question answering system [2], which exploits the information extracted from over 200 million web pages, and went on to win a 2011 *Jeopardy!* television game show against two (human) champions. The work of Microsoft on the Literome project [3] is another impressive realization, where information mined from scientific articles available in PUBMED[1] has been exploited for assisting medical researchers.

Many initiatives have been launched for acquiring large repositories of structured semantic knowledge about our world, including FreeBase [4], YAGO [5], DBpedia [6] or more generally the Linked Open Data [7]. Many such repositories are often collaborative. For instance, DBpedia is built automatically from Wikipedia, which is defi-

[1] http://www.ncbi.nlm.nih.gov/pubmed

nitely a collaborative effort. A few initiatives are (almost) unsupervised, such as the NELL system [8], which continuously learns to extract knowledge from web pages.[2]

While these repositories are continuously growing, they still suffer from two main shortcomings. First, they lack coverage for specialized domains. There does not seem to be many repositories that would be useful for, say, developing a system to answer questions on network protocols. Second, they are mainly English-centric. One might argue that this is not an issue since semantics are not language-specific, but this amounts to an oversimplification. Texts in a given language could very well yield some useful information (or points of view) that are glossed over or simply absent from English documents. More practically, concepts in KBs are associated with English strings (e.g. `ref:label` in DBpedia) that systems can locate in texts, which limits portability to other languages. We are aware of multilingual initiatives, such as BabelNet [9], but their coverage is poor.

This study attempts to tackle some of these shortcomings. We describe a recall-oriented OIE system designed to extract knowledge (triples) from French corpora. We then put it to the test by assessing how general domain knowledge (from French Wikipedia) can be used for deriving new information in domain-specific documents (Érudit, a collection of scholarly papers in the humanities). We believe that the present study is the first one that focusses on such a cross-domain use in OIE. Although a few domain-specific OIE systems have been designed, such as the Literome project aforementioned, they mostly rely on a huge collection of domain-specific texts.

We start by discussing related work in Section 2. We describe in Section 3 our effort to develop an OIE system for the French language. In Sections 4 and 5, we show how it is possible to derive and characterize entity types from triples extracted in Wikipedia and then use these types to classify entity instances found in triples obtained from another corpus. We conclude in Section 6.

2 Related Work

Since the seminal work conducted on TextRunner [10], several toolkits have emerged to allow Open Information Extraction (OIE) in NLP applications. For instance, REVERB [11] relies on part-of-speech tagging and is available for analyzing English text. The WOE extractor [12] was one of the first to propose distant supervision (in their case, they used arguments in Wikipedia infoboxes) in order to mine patterns of interest. Other extractors, such as OLLIE [13], exploit the dependency parse of sentences in order to extract more precise and more diversified relations.

Dealing with facts acquired by such tools on a large corpus is a daunting task. See for instance the impressive effort made at Google [14] for building a huge collection of facts from as many sources as possible (including manually curated databases). Structuring those facts into a useful KB is an even more challenging endeavour, one that has benefited from exciting developments in the recent years.

---

[2] Some supervision was provided at the very beginning of the project in order to identify a number of interesting relations, and there is also human feedback after each iteration of the system in the form of ratings on some newly extracted facts.

Some systems have been designed to learn to structure extraction patterns. For instance, PATTY [15] exploits types (*person*, *location*, etc.) defined for some named entities in manually curated repositories such as YAGO to learn that, for instance, the extraction pattern <*person*> *winner of* <*award*> implies the pattern <*person*> *nominated for* <*award*>. In other systems, the types of entities or relations in texts emerge from the acquired facts thanks to clustering. A convincing example of this is the WEBRE system [16] in which *itemsets* emerge from a collection of untyped facts, such as {*marijuana*, *caffeine*, …}, {*cause*, *result in*, …}, {*insomnia*, *emphysema*, …}.

Such realizations lead to interesting new applications. For instance, in [17], the authors describe a tool primarily aimed at data scientists, allowing them to explore a large collection of documents, thanks to structured extraction patterns. This is useful for rapidly designing a specific extractor, e.g. a tool for mining architects in texts.

Despite these developments, there are still a number of issues that point to a need for better technology, as we show later in this study. First, extraction toolkits are prone to errors, and better extractors must be learned (see for instance [18] for some possible directions). Second, many (if not most) facts acquired are either uninformative and/or anaphoric (e.g. (*she*, *continues*, *her study*)). While anaphoric facts may be partially sanitized by coreference resolution, measuring the informativeness of a fact is still an unresolved issue.

3 Partial Adaptation of REVERB to French

The original REVERB Open IE system [11] proceeds in two phases, both of which are language-dependent.

1. The first stage reads POS-tagged and NP-chunked sentences and produces a (possibly empty) set of extraction triples (*arg1*, *r*, *arg2*), e.g. (*President Obama, gave a talk at, the White House*). To be extracted, a triple must satisfy a few constraints. A syntactic constraint stipulates that a relation *r* must match a regular expression based on parts of speech. A lexical constraint learned on a large corpus removes overspecified relations (e.g. *are only interested in part of the solution for*). If a relation is located, noun phrases to its right and left must also be present and valid.
2. A second stage uses a logistic regression classifier in order to filter out dubious or uninformative triples. REVERB's authors selected 19 features that are used to build this confidence function crucial to weed out uninformative and incoherent extractions (at the cost of recall).

We wanted to take advantage of the extraction engine already provided by REVERB for our study. In our case, however, we did not implement the French equivalents of the lexical constraint and the classifier. We conjectured that the filtering nature of many of the manipulations made on these French triples in this experiment would naturally eliminate some of the noise.

Thanks to the high quality of its API, REVERB lends itself very well to a "translation" in another language. For French, we had to make the following modifications.

- The preprocessing steps relying on Apache OpenNLP were adapted to use French statistical models for sentence segmentation, word tokenization, part-of-speech tagging and noun-phrase chunking. These models are trained on a large generic corpus [19] and are freely available on the web[3] for the OpenNLP framework.
- The empirical POS-based regular expression at the heart of relation extraction was changed to the one shown in Fig. 1, which is the result of our own attempts to capture as many relations as possible on a small development set. Aside from the fact that the French POS tagset differs slightly from the English one, a noteworthy difference is the presence of clitics in this pattern. Indeed, clitics frequently occur in French verb phrases, e.g. *ils s'y sont revus* (*they saw each other there*), whose POS tags are CLI (*ils*) CLI (*s'*) CLI (*y*) V (*sont*) *PP* (*revus*). Another distinction is the absence of nouns and determinants in the expression, which greatly reduces the need for the aforementioned lexical constraint present in the English REVERB, at the cost of missing some French verb phrases containing these elements (e.g. *faire partie de – be part of*). We felt that the added precision was worth this alteration.

ADV? CLI* V PP? ADV? PP? INF? ADV? PREP?
INF = (VINF | PREP VINF)
ADV = *adverb*, CLI = *clitic*, V = *verb*, PP = *past participle*, VINF = *infinitive*

Fig. 1. Regular expression used in the REVERB extraction engine for French. The special symbols ? and * indicate respectively "once or not at all" and "zero or more times".

Dans la première partie de Surveiller et punir, Michel Foucault décrit le rôle politique et social du supplice durant l'époque qui précède les réformes pénales de l'âge classique.
(*Michel Foucault, décrit, le rôle politique*)
(*l'époque, précède, les réformes pénales de l'âge classique*)

Fig. 2. A sample sentence extracted from the `erudit` corpus yields two extracted triples.

4 Triple Extraction

4.1 Corpora

We extracted triples from two distinct French corpora, called `wiki` and `erudit`. The `wiki` corpus is a text serialization of all French Wikipedia articles as of June 2014 – 1.5 million articles, in total[4]. The `erudit` corpus is derived from the online collection of scholarly and cultural journals curated by the Érudit Consortium, which consists of 158 journals, mostly in the humanities[5]. We extracted the raw text from 19k XML documents in French, a task made easy by the principled tagging effort carried out on these documents by the Érudit team.

[3] `sites.google.com/site/nicolashernandez/resources/opennlp`
[4] `http://rali.iro.umontreal.ca/rali/en/wikipedia-dump`
[5] `http://erudit.org/revue/`

The statistics for these corpora are shown in Table 1. It is worth noting that, beside their dissimilar domains, the two corpora also differ sharply in their nature by their average sentence length, most likely due to the comparative verbosity of scholarly papers that make up the `erudit` corpus.

Table 1. Corpus statistics for the Wikipedia corpus and the corpus derived from Érudit.

Corpus	Domain	Docs	Sentences	Tokens	Forms	Tokens/Sent
`wiki`	generic	1.5M	31.1M	668M	28.7M	20
`erudit`	humanities	19k	2.8M	96M	2.8M	34

4.2 Extraction of Triples

We carried out the extraction of triples (*arg1, r, arg2*) using the modified version of REVERB described in Section 3. We completed the extraction process by lemmatizing the verbs present in *r*. Therefore, the following discussion only concerns verbs in their infinitive form. Table 2 shows the extraction statistics.

Table 2. Extraction statistics for corpora `wiki` and `erudit`. We only use triples without pronouns in this study, losing about a third of the original triples. The remaining statistics indicate the number of different relations, arg1s and arg2s found in these filtered triplets.

Corpus	Raw triples	Triples w/o pronouns	Relations	arg1s	arg2s
`wiki`	30.4M	20.8M	1.2M	7.2M	7.4M
`erudit`	4.7M	3.1M	0.4M	1.3M	1.4M

The triples in both corpora follow a Zipfian distribution. For `wiki`, the frequency of a triple can be approximated by $freq = 10453 \times rank^{-0.716}$, with $R^2 = 0.997$, although the three most frequent triples are overrepresented. They have a frequency of about 30k each and are the product of a Wikipedia template on demographics. The most frequent reads (*The evolution in population, is known, throughout*).

The most frequently occurring relations in corpus `erudit` are *être*, *avoir*, *faire*, *devenir* (*be, have, do, become*), involved in 17% of all triples extracted. The arguments *arg1* and *arg2* are dominated by pronouns. For both corpora, the ten most frequent *arg1*s are all pronouns. Since these anaphora render their triplets uninformative, we filtered them using a simple blacklist, losing a third of the raw triples.

5 Classification of Entity Instances

In this study, we attempt to classify **entity instances** found in extracted triples (e.g. *Michel Foucault*) into **entity types** (e.g. *auteur – author*). Entity instances are not limited to traditional named entities. For example, we would also like to classify an *article* as an *étude* (*scientific study*). An additional challenge stems from the fact

that we use the large `wiki` corpus to define the entity types and then proceed to classify instances found in triples extracted from the `erudit` corpus.

5.1 Selection of Entity Types

We start by defining a set of entity types in a loosely supervised way. We filter the triples (*arg1*, *r*, *arg2*) from `wiki` by keeping only those that satisfy the constraints that *r* must be the verb *être* (*to be*) and *arg2* must contain a common noun *t*. All such elements *t* whose frequency is greater than an empirically devised threshold of 1000 are kept and constitute the set *T* of entity types considered in this study.

The set *T* contains 358 types. The five most frequent types are *commune* (*municipality*), *espèce* (*species*), *village*, *film* and *ville* (*city*), presumably reflecting the relatively large number of Wikipedia articles devoted to these topics. Some topics have overlapping meanings, like *commune* and *ville*. Other such overlapping topics comprise *philosophe* (*philosopher*), *auteur* (*author*) and *écrivain* (*writer*). We did not regroup or filter out these overlapping categories. On a related note, we consider entity types as non-hierarchical: even when a type t_1 semantically subsumes a type t_2, they are considered completely distinct.

5.2 Characterizing Entity Types by Their Relations

Once the set *T* of entity types is built, we characterize each $t \in T$ by the relations where *t* is involved, reasoning that different instances of a given type *t* must have similar relations, and that these relations should differ from those involved with instances of a different type.

We therefore seek to build a **relation profile** P_t for each *t*. At its simplest, this profile will include relations (i.e. verbs) and their associated count. We find the relations of interest in the triples extracted from the `wiki` corpus.

We start by identifying a set of instances for each *t*. We gather all *arg1*s from triples (*arg1*, *r*, *arg2*) where *r* is the verb *être*, *arg2* contains the common noun *t*, and *arg2* is not a hapax. This works reasonably well, but instances are often contaminated with ubiquitous instances. For example, a third of all 358 topics contain the instance *son père* (*his father*). Manual examination revealed that anaphora is to blame for this phenomenon, since these pervasive instances are associated with multiple types (as in *his father was a physicist* and *his father was a sportsman*). To compensate for this, we removed entity instances appearing in more than 2% of the 358 topics, an upper limit we deemed "reasonable" on the number of types an instance can belong to. There are 237 instances per type on average (min: 1, max: 4953). Table 3 shows a random sample of the instances identified in the `wiki` corpus for the types *auteur* (*author*) and *période* (*period of time*). A manual evaluation of half a dozen instance lists reveals a 90% precision. Errors vary from slight inaccuracies in classification (e.g. while *Jean-Florian Collin* has written a few books, he is primarily known as an architect and politician, not as an *author*) to flagrant extraction issues (e.g. *le contexte des noms de domaine* (*domain name context*) is not a *period of time*).

Table 3. Instances for two types: *auteur* (*author*) and *période* (*period of time*).

Instances for *author*	**Instances for *period of time***
Farid Boudjellal	l'estive (*grazing period*)
Jean-Florian Collin	1890 (*1890*)
Richard Matheson	le contexte des noms de domaine (*domain name context*)
Bernard Yaméogo	la première restauration (*First Restoration*)
Hanns Heinz Ewers	le chalcolithique (*Copper Age*)
… 416 instances overall	… 152 instances overall

From the relatively precise list of instances for each type *t*, we are able to inspect triples containing these instances at the *arg1* position and gather all corresponding relations in order to build a relation profile P_t.

For the entity type *auteur* (*author*), the 10 most frequent relations gathered are *be*, *do*, *have*, *write*, *become*, *give*, *emeritus*, *take*, *put*, *run* and *say*.[6] A systematic error during part-of-speech tagging gives rise to the erroneous *emeritus*, where the French word *émérite* is mislabeled as a verb rather than as an adjective. For *période* (*period of time*), the most frequent relations are: *be*, *mark*, *see*, *follow*, *have*, *do*, *become*, *put*, *av* and *know*. Here, *av* is also due to an unfortunate tagging error.

We distinctly face a situation where uninformative relations (*be*, *do*, *have*, etc.) appear in all profiles. At the same time, more type-specific relations emerge (*write*, *say*, for an *author*; *mark* and *follow* for a *period of time*).

To get a better sense of the quality of these relation profiles[7], we add a tf–idf score to each relation in a given profile P_t. We computed tf–idf by considering each profile as a document, populated with words that correspond to the relations. In other words, for each relation in a given P_t, tf is the relation count and idf is proportional to the inverse of the number of profiles containing the relation. Table 4 shows an excerpt of the profile for the entity type *auteur* (*author*).

5.3 Extracting Entity Instances from `erudit` to Create Dev and Test Sets

Our goal is to label entity instances with the correct entity type. While these types were extracted from the corpus `wiki`, we select instances from the corpus `erudit`, in an effort to assess how well the entity types from one corpus generalize to another one, with a different writing style (see Section 4.1).

We performed triple extraction with the adapted REVERB previously described on `erudit`. We then selected all *arg1*s whose frequency is greater than 50. We extracted their relation profiles as explained in Section 5.2.

To create a data set, we manually labeled the 120 instances featuring the most relations in their relation profile. We discarded instances that were ambiguous (e.g. the last name *Tremblay* is not enough to identify the entity), were the result of extraction

[6] We translate the French relations for the sake of clarity.

[7] We also compute these scores for classification purposes (see Section 5.3).

Table 4. Relation profile excerpt for the entity type *auteur* (*author*). For each relation, a translation is provided for convenience. Relations are listed in decreasing order of tf–idf score. The relation *gothique* (*gothic*) is due to a POS-tagging error.

author	
Relation	**Frequency**
run	293
diverge	38
report	197
speak	239
author	95
write	635
969 relations	∑ = **33671**

period of time	
Relation	**Frequency**
gothic	364
happen	220
mark	1709
start	218
follow	1405
change	95
1029 relations	∑ = **41954**

errors (e.g. an adjective mislabeled as a noun) or simply did not belong to any types (e.g. *snow*, *orientation*). Each instance received a label consisting of the entity types extracted in Section 5.2 to which it belongs. For example, *Michel Foucault* received the labels *philosopher*, *author*, *writer* (2.85 labels per instance on average).

The classification algorithms described in Section 5.4 were tuned on 20 of these instances while the other 100 constituted the test set. We did not need a large number of development instances, since our classifiers do not require the production of a model. The test set was used to assess the impact of a few parameters and to fine-tune some of them.

5.4 Classifiers

Our goal is to classify entity instances (e.g. *Michel Foucault*, *article*) extracted from `erudit` into entity types (e.g. *author*, *scientific study*) characterized by the relation profile observed in the corpus `wiki`. We tried 4 different approaches to find the closest relation profile for a given instance. Let $P_t = \{r_{t1}, r_{t2}, \dots, r_{tm}\}$ be the set of all m relations for a given type t. Let $P_i = \{r_{i1}, r_{i2}, \dots, r_{in}\}$ be the set of all n relations for a given instance i. Let sim be a similarity function between two profiles. We attribute a type t^* to a given instance i by finding $t^* = \underset{t}{\operatorname{argmax}}\, \mathrm{sim}(P_t, P_i)$. The set of all possible types t includes all the types manually identified during labeling (34 types) to which we added 16 other types randomly selected from the set T of all 358 types identified in Section 5.2, for a total of 50 possible types available to each classifier.

For each similarity metric, we added a parameter λ that constrains the number of relations considered in each profile when computing a similarity. The value λ specifies that only the λ-most frequent relations in P_t and P_i should be used. The others are ignored. This allows the algorithm to focus on the most represented relations in each profile. The optimal λ for a given similarity metric is found by exhaustive search on the development dataset. We also investigated results without any such filter.

jaccard is our baseline and consists in computing the generalized Jaccard similarity coefficient between the two relation profiles. We first derive a frequency vector for each profile. For example, for P_t, we obtain $\mathbf{v}_t = \langle freq(r_{t1}), freq(r_{t2}), \dots \rangle$, where

$freq(r_{ti})$ is the count of the relation r_i, for the profile *t*. The similarity can then be computed using the following formula:

$$jaccard(\mathbf{v}_t, \mathbf{v}_i) = \frac{\sum_j \min(\mathbf{v}_t[j], \mathbf{v}_i[j])}{\sum_j \max(\mathbf{v}_t[j], \mathbf{v}_i[j])}$$

cos is the cosine similarity between two profiles, and is computed using the frequency vectors $\mathbf{v}_t$ and $\mathbf{v}_i$ discussed above. We also tried a variant **cos-bin** where, instead of the frequencies of the relations, the vectors are encoded with 1 (the relation is present in profile) or 0 (the relation is absent).

tfidf makes use of the tf–idf scores we introduced in Section 5.2. The similarity function here is comparable to the information retrieval scenario where the relations in P_i constitute the query and the different P_t are each a document in a collection. The "most relevant" P_t is therefore the most similar. The similarity score for a given P_i is the sum of the tf–idf scores for each relation found in P_t.

Finally, **kl** is a comparison of the relation distributions in P_t and P_i using the Kullback–Leibler divergence $D_{\mathrm{KL}}(Q_t \parallel Q_i)$, where Q_t and Q_i are the probability distributions (relative frequencies) of relations in P_t and P_i respectively. KL divergence does not handle 0s in these distributions, so we smooth by replacing them by a small value ε whose value was tuned on the development set. We also experimented by throwing out the dimensions with 0 values, with disappointing results (not presented here).

5.5 Results and Error Analysis

The classification errors for all similarity metrics are presented in Table 5. An instance is considered misclassified if the closest type *t** returned by a given similarity metric is not part of the labels attributed manually. The results are unsatisfactory for all metrics, except the KL divergence, with a classification error rate of 34% on the test set, lower than random (92%) and the Jaccard baseline (55%). The metrics generalize quite well from the development set to the test set.

Table 5. Classification results for 5 similarity metrics on the development set and the test set. The **kl** algorithm (Kullback–Leibler divergence) yields the best results (in gray). The **random** classifier picks a solution at random while **most frequent** always picks the most frequent label (*city*). The latter two are provided for comparison.

Similarity metric	**Classification error (%)**	
	dev (*n* = 20)	**test** (*n* = 100)
random	90%	92%
most frequent (*city*)	90%	91%
jaccard (λ = 50)	55%	55%
cos (λ = 50)	60%	64%
cos-bin (λ = 50)	55%	58%
tfidf (λ = 500)	50%	53%
kl (λ = 100)	30%	34%

The two authors of this study manually inspected the results for **kl** to identify the kinds of errors the similarity function led to. We identify two kinds of errors: "hard" and "soft" errors (for lack of better terms). Hard errors occur when an entity is unambiguously mislabeled, e.g. *Socrates* is a *municipality*. Soft errors arise when the predicted type is not entirely incompatible with the instance to label, e.g. *United States* is a *group*. This is admittedly a subjective exercise, however the reader can judge for himself by examining a sample of this partition in Table 6. Overall, out of 34 errors, we found that 20 were hard and 14 were soft. Had we tolerated the latter, the classification error for **kl** on the test set would have fallen to 20%. These 20 hard errors contain 10 misclassifications where a country (*Canada*, *Mexico*) is classified as a *city*.

Table 6. A sample of a few instances, their manual labels and the predicted type by the kl algorithm. "Hard" classification errors are noted with a double dagger (‡), "soft" ones with a †.

Instance	**Type labels**	**Predicted type** (kl)
1960	*year, period of time*	*period of time*
Aragon	*artist, author, writer*	*author*
article	*scientific study, book, compilation*	*scientific study*
Belgium	*country, place, places, toponym*	*city*‡
French Canadians	*people, group*	*characters*†
Germany	*country, place, places, toponym*	*city*‡
Health Canada	*organisation, association, organism, group*	*physician*†
Michel Foucault	*philosopher, writer, author*	*author*
prime minister	*minister, president, master*	*minister*
Socrates	*philosopher, writer, author*	*city*‡
text	*scientific study, book, compilation*	*book*

The parameter λ significantly affects the performance of the kl similarity metric. There seems to be an optimal value in the interval [50, 1000] at around 40% classification error, outside of which the performance is poor. Both the development and test set exhibit the same behavior.

6 Discussion

One of the goals of this paper was to attempt open information extraction (OIE) in French and assess the difficulties encountered while doing so. The adaptation of REVERB went smoothly, partly because there are drop-in French replacements for the POS and chunking statistical models the software uses, and partly because its API is expertly written. We also opted not to adapt all of REVERB's filters to French, because we favoured recall over precision in our architecture. However, we feel that implementing the rest would be straightforward should we need it in the future.

Like most OIE approaches, the problem of uninformative and ambiguous triples is significant. We lose a third of extracted triples to pronominal anaphora alone, which

amounts to 10M triples for the corpus `wiki`. This highlights the need for a robust anaphora resolver. For French, a recent study on a commercial-grade grammar checker [20] shows that 70% of these anaphora could be resolved successfully, a possible addition of 7M triples of information in our case. Naturally, the figure of 10M triples lost is a minimum, since it does not take into account other types of anaphora (e.g. *his father*). However, our system behaved reasonably well in the face of these latter problems, thanks to simple frequency thresholds akin to idf (inverse document frequency), reasoning that ubiquitous instances are bound to be non-specific and uninformative.

The second goal of this paper was to explore whether it was possible to extract information from a generic corpus (`wiki`) and use it to infer new knowledge in a different, domain-specific corpus (`erudit`) through the analysis of OIE's resulting triples. We showed that it is indeed possible to identify and characterize entity types by the relations their respective instances are associated with. It then becomes possible to put these profiles to good use and classify instances extracted from the other corpus, for two thirds of these instances. To our knowledge, in this context, this approach is original. It does suffer however from the fact that the instances to classify must be relatively frequent (in order to gather enough information on them). The system described here would be hard-pressed to associate a hapax instance to an entity type, for instance. Moreover, establishing "relation profiles" proves sensitive to systematic extraction errors, notably those committed during part-of-speech tagging. A tagging error that mislabels an adjective for a verb in a specific context (like *gothic* preceded by *author*) is bound to create significant artefacts in relation profiles, since the latter are designed to gather just such systematic specificities, whether they are linguistically motivated or the result of an extraction problem.

There is room for improvement when considering the figure of 34% of classification error reported here. We identify some possible solutions above. However, the same statistic also shows that there is definite potential in the idea of exploiting the knowledge derived by OIE from a generic corpus and then applying it to a stylistically and thematically different collection of texts.

References

1. Sowa, J.F.: The Challenge of Knowledge Soup. In: epiSTEME-1. pp. 55–90 (2004).
2. Ferrucci, D.A.: IBM's Watson/DeepQA. In: Proceedings of the 38th Annual International Symposium on Computer Architecture. ACM (2011).
3. Poon, H., Quirk, C., DeZiel, C., Heckerman, D.: Literome: PubMed-Scale Genomic Knowledge Base in the Cloud. Bioinformatics. 30, 2840–2842 (2014).
4. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. pp. 1247–1250 (2008).
5. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: A Core of Semantic Knowledge. In: 16th international World Wide Web conference (WWW 2007) (2007).

6. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Semantic Web Journal. 6, 167–195 (2015).
7. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. 5, 1–22 (2009).
8. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr, E.R.H., Mitchell, T.M.: Toward an Architecture for Never-Ending Language Learning. In: Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010) (2010).
9. Navigli, R., Ponzetto, S.P.: BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial Intelligence. 193, 217–250 (2012).
10. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open Information Extraction from the Web. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence. pp. 2670–2676. , Hyderabad, India (2007).
11. Fader, A., Soderland, S., Etzioni, O.: Identifying Relations for Open Information Extraction. In: Empirical Methods in Natural Language Processing. pp. 1535–1545 (2011).
12. Wu, F., Weld, D.S.: Open Information Extraction Using Wikipedia. In: 48th Annual Meeting of the Association for Computational Linguistics. pp. 118–127 (2010).
13. Mausam, Schmitz, M., Bart, R., Soderland, S., Etzioni, O.: Open Language Learning for Information Extraction. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 523–534 (2012).
14. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 601–610. ACM (2014).
15. Nakashole, N., Weikum, G., Suchanek, F.: PATTY: A Taxonomy of Relational Patterns with Semantic Types. In: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1135–1145 (2012).
16. Min, B., Shi, S., Grishman, R., Lin, C.: Ensemble Semantics for Large-scale Unsupervised Relation Extraction. In: EMNLP and CoNLL. pp. 1027–1037 (2012).
17. Akbik, A., Visengeriyeva, L., Kirschnick, J., Löser, A.: Effective Selectional Restrictions for Unsupervised Relation Extraction. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing. pp. 1312–1320 (2013).
18. Koch, M., Gilmer, J., Soderland, S., Weld, S.D.: Type-Aware Distantly Supervised Relation Extraction with Linked Arguments. In: Empirical Methods in Natural Language Processing (EMNLP). pp. 1891–1901 (2014).
19. Hernandez, N., Boudin, F.: Construction automatique d'un large corpus libre annoté morpho-syntaxiquement en français. In: Traitement Automatique des Langues Naturelles (TALN). , Sables d'Olonne, France (2013).
20. Pironneau, M., Brunelle, É., Charest, S.: Pronoun Anaphora Resolution for Automatic Correction of Grammatical Errors (Correction automatique par résolution d'anaphores pronominales) [in French]. Proceedings of TALN 2014 (Volume 1: Long Papers). 1, 113–124 (2014).