

Evaluation of Unsupervised Information Extraction

Wei Wang*, Romaric Besançon*, Olivier Ferret*, Brigitte Grau†

*CEA, LIST, Vision and Content Engineering Laboratory
F-91191 Gif-sur-Yvette, France
{wei.wang,romaric.besancon,oliver.ferret}@cea.fr

†LIMSI-CNRS
BP 133
F-91403 Orsay Cedex, France
brigitte.grau@limsi.fr

Abstract

Unsupervised methods gain more and more attention nowadays in information extraction area, which allows to design more open extraction systems. In the domain of unsupervised information extraction, clustering methods are of particular importance. However, evaluating the results of clustering remains difficult at a large scale, especially in the absence of a reliable reference. On the basis of our experiments on unsupervised relation extraction, we first discuss in this article how to evaluate clustering quality without a reference by relying on internal measures. Then we propose a method, supported by a dedicated annotation tool, for building a set of reference clusters of relations from a corpus. Moreover, we apply it to our experimental framework and illustrate in this way how to build a significant reference for unsupervised relation extraction, more precisely made of 80 clusters gathering more than 4,000 relation instances, in a short time. Finally, we present how such reference is exploited for the evaluation of clustering with external measures and analyze the results of the application of these measures to the clusters of relations produced by our unsupervised relation extraction system.

Keywords: unsupervised information extraction, relation extraction, clustering evaluation

1. Introduction

Evaluation is still a problematic issue in the Natural Language Processing field, especially when unsupervised approaches are used. Nevertheless, unsupervised methods in information extraction area gain more and more importance to deal with the large amount of information from digital resources, notably from the Internet. They avoid some shortcomings of supervised or semi-supervised methods, such as the need to define statically the type of relations to focus on or the need to annotate a significant number of examples or to provide a significant number of seeds. Initially defined in the context of traditional information extraction with work on *On-demand information extraction* (Hasegawa et al., 2004) or *Preemptive Information Extraction* (Shinyama and Sekine, 2006), unsupervised information extraction has also taken the form of *Open Information Extraction* (Banko et al., 2007) when it was applied for large-scale knowledge acquisition from the Web. All these forms of unsupervised information extraction share two main tasks:

- extracting relations between entities from texts without fixing *a priori* their type;
- clustering similar extracted relations to characterize their type.

As a consequence, clustering methods and their evaluation are particularly important for unsupervised information extraction approaches. Although the development of new clustering methods is an active field, their evaluation is still a challenging problem, especially for domains in

which large reference data do not already exist. Some works use unsupervised information extraction as a source of improvement for “traditional” information extraction by extending the coverage of models learned from annotated corpora. In this perspective, unsupervised information extraction modules are indirectly evaluated through their impact on the information extraction system they are part of, as in (Banko and Etzioni, 2008) or (González and Turmo, 2009).

Our viewpoint is different since we use unsupervised information extraction as a means to draw a global picture of the relations between a set of target entities for technology watch purposes. Hence, we are interested in evaluating more directly the clusters of relations built by this kind of processes. In this article, we tackle more precisely this issue from the two following viewpoints:

- how to evaluate clustering results without any reference?
- how to build a reliable reference for a given corpus and use it for evaluation?

The first viewpoint is new in the field of unsupervised information extraction while the second one arises from the analysis of existing work. (Hasegawa et al., 2004), one of the first work in this domain, evaluated *a posteriori* clusters of relations by assigning manually to each of them the relation type corresponding to the majority of the relations this cluster contains. Then, recall and precision measures were computed by counting pairs of relations that were correctly or not part of the same cluster. However, such *a posteriori*

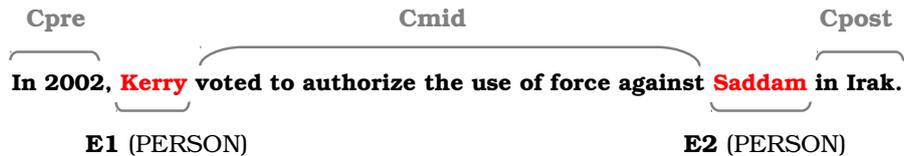


Figure 1: Example of extracted relation

approach faces two problems, that are linked: first, because of its cost, an evaluation cannot be done each time a new clustering system or an existing clustering system with different parameters is tested; second, the results of the evaluation of one system cannot be used for the evaluation of another one as the reference built from the first evaluation is biased by the first system. This difficulty could be overcome to some extent by using a pooling technique, as it is often done for the evaluation of search engines. However, pooling requires having a large number of different systems, which is only possible in the context of an evaluation campaign, and is made more difficult in the case of clustering by the fact that results are not structured by a set of known queries and are therefore more difficult to compare.

(Rosenfeld and Feldman, 2007) adopted a different approach more directly linked to our viewpoint. First, they annotated manually a restricted set of 200 relations and then, computed the *Jaccard coefficient* between their clustering results and their reference clusters at the level of relation pairs. The size of their reference set of relations was however small and we propose in this article both the methodology and the tools for building manually a large set of reference clusters of relations from a corpus and using it for evaluating an unsupervised information extraction system. This system and more globally the framework of our experiments is presented in Section 2. Section 3 defines how internal clustering measures can be applied to unsupervised information extraction while Section 4 details our approach for its external evaluation, using a reference corpus.

2. Unsupervised extraction of relations

In the context of unsupervised information extraction, a flexible scheme has to be defined for the relations to extract that does not require the definition of their type. Our prototype of relation candidates is characterized by two main information, as shown in the example of Figure 1:

- a pair of named entities ($E1$ and $E2$);
- the linguistic form of the relation, which is made of the three parts of the sentence from which the relation is extracted: before $E1$ ($Cpre$), between $E1$ and $E2$ ($Cmid$), after $E2$ ($Cpost$).

The extraction of such kind of relations was applied in an open domain perspective to a sub-part of the AQUAINT-2 corpus made of 18 months of the *New York Times* newspaper. The whole relation extraction process consists more precisely of four tasks:

Linguistic preprocessing OpenNLP¹ tools were used to analyze documents to obtain linguistic information such as named entities, parts-of-speech and normalized words.

Candidate extraction For each sentence of the corpus, all named entity pairs are extracted with the only constraint that at least one verb must exist between the two entities. Six types of pairs are considered in this experiment. Table 1 gives their volume for our corpus.

Relation type	Initial extraction	Post-filtering
ORG – LOC	71,858	15,226 (21%)
ORG – ORG	77,025	13,704 (18%)
ORG – PER	73,895	10,054 (14%)
PER – LOC	152,514	47,700 (31%)
PER – ORG	126,281	40,238 (32%)
PER – PER	175,802	38,786 (22%)

Table 1: Volume of extracted candidates

Relation filtering We observed in practice from this initial extraction that many candidate relations do not refer to a true relation between their named entities. Hence, a relation filtering step was applied to remove as much as possible these false instances. About 25% of initial relations (also shown in Table 1) were kept after two steps of filtering: the first one by heuristics to get rid efficiently of highly probable false relations resulting from indirect speech or very long sentences; the second one by a machine learning model trained to filter more finely candidate relations, using a strategy similar to (Banko and Etzioni, 2008). Our best statistical model, a linear Conditional Random Fields model (CRF), achieves 0.762 as precision and 0.782 as recall (Wang et al., 2011).

Relation clustering Extracted candidates are clustered to group similar relations together in order to offer a better view of existing relations between named entities. This step relies on both a similarity measure and a clustering algorithm. For the former, we chose the *cosine* measure, applied to a bag-of-words on the $Cmid$ part of relations. The *All Pairs Similarity Search* (APSS) (Bayardo et al., 2007) algorithm was used to compute efficiently a similarity matrix of all filtered relations whose similarity is higher than a fixed threshold, which avoids the computation of all pairwise similarities. For the latter, the Markov Clustering algorithm (MCL) (van Dongen, 2000) was applied to create clusters of relations according to similarities computed with APSS. This algorithm does not require to fix a predefined number of clusters, which would not be possible in our case. Moreover, MCL converges quickly in practice

¹<http://opennlp.sourceforge.net>

	<i>Expected density</i>		<i>Connectivity (p = 20)</i>	
	pre-filtering	post-filtering	pre-filtering	post-filtering
ORG – ORG	1.06	1.13	5335.7	3450.8
ORG – LOC	1.13	1.02	4458.7	2837.6
ORG – PER	1.09	1.17	3025.4	1532.4
PER – ORG	1.02	1.06	5638.0	4620.0
PER – LOC	1.08	1.07	5632.5	4571.3
PER – PER	1.13	1.15	3892.7	2569.2

Table 2: Internal evaluation for relation clustering (best results are presented in bold)

through a series of random walks performed on a similarity graph directly built from the similarity matrix. This clustering procedure was carried out on both relations after the relation filtering step and relation candidates before this step for evaluating with different kinds of measures the interest of the filtering of extracted relations.

3. Internal evaluation of clustering quality

When no reference is available, clustering quality is usually evaluated by a manual inspection of a subset of clustering results, which is likely to be biased as the resulting clusters tend to influence annotators. Hence, we explored a new approach in the field of unsupervised relation extraction through the use of internal criteria. Such criteria establish to which extent the clusters obtained are representative of similarity values between relations (Halkidi et al., 2002). (Carugo, 2010) proposed to estimate clustering tendency² by computing the *Hopkins Coefficient* or the *Cox-Lewis Coefficient* for random selected sets in clustering results. Other internal measures include *Dunn Index* family measures, *Davies-Bouldin Index* and *expected density*. We first chose *expected density* since it was proved to have the best and the more stable correlation with F-measure for document clustering, especially compared to the more widespread *Dunn Index* (Stein et al., 2003).

Given a weighted graph (V, E, w) with a node set V , an edge set E and a weight function w , the density θ of the graph is defined by:

$$\theta = \frac{\ln(w(G))}{\ln(|V|)}$$

with $w(G) = |V| + \sum_{e \in E} w(e)$, and the weight function w is defined by the relation similarity in our case.

Expected density can be computed by local and global graph density of clustering. For a set of result clusters $C = \{C_i\}$ with $C_i = (V_i, E_i, w)$, the expected density can be defined by :

$$\rho = \sum_{i=1}^{|C|} \frac{|V_i| \theta_i}{|V| \theta}$$

where $\frac{|V_i|}{|V|}$ intends to balance the different size of each cluster. The higher value of measure ρ implies better clustering quality.

²Clustering tendency tries to determine if applying clustering is likely to produce interesting results or not. It can also refers to the estimation of the number of clusters before clustering.

We also considered the *Connectivity* measure (Handl et al., 2005), another internal measure that evaluates how many nearest neighbors, according to the similarity matrix, are not clustered together. This measure is of particular interest since it is based on the similarity graph of our clustering method. The connectivity measure is defined by:

$$c = \sum_{i=1}^{|V|} \sum_{j=1}^p x_{i,nn_i(j)}$$

with p denotes the number of neighbors considered, $nn_i(j)$ is the j^{th} nearest neighbor of i , and $x_{i,nn_i(j)}$ equals to 0 if i and $nn_i(j)$ are in the same cluster, and equals to 1 otherwise.

A random subset of the total corpus (5,000 relations in our experiments) was selected for calculating connectivity to neutralize the dependence of this measure on the size of the input graph. This measure is inverse compared to the expected density: a lower connectivity value indicates a better clustering.

Results of expected density and connectivity measures are presented in Table 2 and show the positive impact of the filtering of relations: clusters built from post-filtered relations are generally better compared to clusters built from pre-filtered relations with the same clustering method. The two entity pairs that do not follow the same tendency are, for the expected density, ORG–LOC and PER–LOC. Since both share the same entity type *location*, this observation probably indicates a special behavior of these entities. More precisely, location entities are often included in adverbial phrases, in which case there is no real relation between the location entity and the other entity. However, with the current similarity measure, phrases with similar location adverbials can be clustered together and obtain a good clustering score.

4. Clustering evaluation with reference

The reference for clustering evaluation must be carefully constructed in a way that integrates the following three considerations:

- having a large number of clusters with a significant size, in order to make the evaluation representative;
- having a large variety of the expressions of a relation in a cluster, in order to take into account several ways of expressing the relation that are semantically equivalent (paraphrases) and have a richer and more realistic reference;

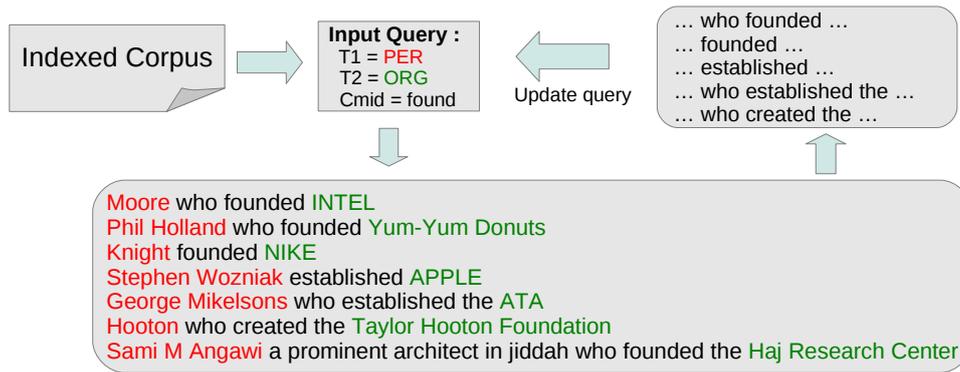


Figure 2: An example of bootstrapping for building reference clusters

- having a balanced representation of each expression of a relation in a cluster, in order to avoid potential biases: some expressions may be a lot more frequent than others, but we do not want their contribution to the evaluation to be predominant, so that we can really evaluate the capacity of the clustering to group different expressions of a relation.

In this section, we first present the method we used to build a reference set of relation clusters for a given corpus. Then, we evaluate the clusters of relations of Section 2 against this reference and analyze the results of this evaluation.

4.1. Building reference clusters

The number of relations extracted from a corpus is generally large. Hence, building reference clusters of relations starts with the indexing of these relations by a search engine (*Lucene* in our case). This indexing takes distinctly into account the components of a relation, in order to let the annotators query them specifically: the named entities ($E1$ and $E2$), the named entity types ($T1$ and $T2$) and the linguistic characterization of the relation ($Cmid$) (see Figure 1). The following bootstrapping procedure is then applied, relying on indexed relations:

1. query the indexed relations by setting one or more fields among $T1$, $T2$, $E1$, $E2$ and $Cmid$;
2. rank resulting relations following the decreasing frequency of their expression ($Cmid$ part);
3. enlarge the set of the most frequent relations by updating the initial query with their characteristics.

As illustrated in Figure 2, for example, given an initial input query $T1 = \text{PER}$, $T2 = \text{ORG}$, $Cmid = \text{found}$, a list of relations with PER-ORG named entity couples are retrieved and ranked according to the frequency of their expression ($Cmid$). These relations can then be added to an existing cluster or used to create a new cluster. Then, more relations can be explored by querying all relations with these named entity couples (e.g. $E1 = \text{Moore}$, $E2 = \text{Intel}$, etc). The results of these new queries are ranked once again according to the frequency of their $Cmid$ part and some $Cmid$ in results can be chosen to update the query for the next iteration. It must be noted that all fields in results can be

used to create new queries in the bootstrapping procedure (e.g. $T1 = \text{PER}$, $Cmid = \text{establish}$, $E2 = \text{Apple}$, or $T1 = \text{PER}$, $T2 = \text{ORG}$, $Cmid = \text{who found}$, etc). Moreover, to obtain the first list of named entity couples, the initial query can be replaced by a knowledge base, that is to say a list of couples of entities as those that can be typically extracted from the *InfoBoxes* of Wikipedia.

With the help of the search engine, the size of clusters can be easily enlarged, especially for high-frequency relations. However, we chose to restrict the number of relations of each cluster so that evaluations will not be dominated by large clusters of too similar relations. Hence, we limit the relations with the same expression to 30 examples in practice. The bootstrapping procedure also takes into account the consideration of variety in the content of clusters as the named entity couples collected after the initial query are used in the exploration step for finding various forms of expression of a relation.

Figure 3 gives an overview of the Web-based annotation tool we developed for supporting this procedure. This tool enables annotators to query the indexed set of relations, to view the relations retrieved by their queries but also to group similar relations together, to add new relations to an existing reference cluster and to visualize its content. Until now, our reference is made of 80 clusters of 4,420 relations. About a dozen of clusters were constructed for each couple of named entity types with sizes varying from 4 to 280 relations. The last column on the right of Figure 3 shows more precisely the number of clusters and the number of relations for each relation type.

4.2. Evaluation with external measures

Table 3 provides a first comparison with our reference by showing how many relation instances from the reference are grouped by our clustering algorithm (i.e. are contained in a cluster of size ≥ 2) both before and after the filtering step. These results confirm the global trend of the evaluation with internal measures: similar relations are more likely to be grouped after relation filtering than before.

For a more comprehensive evaluation, external measures like *Purity*, *Normalized Mutual Information* and *F-measure* have been well discussed in the literature (Manning et al., 2008). Given reference clusters with N relations, all pairs of relations in result clusters can be compared to those

Relation Annotation

Query Fields
 Entity 1 : Cmid: Entity 2 :

Please choose a Knowledge Base File :

Available References :

organization-location :	<input type="radio"/> announce_in(13) <input type="radio"/> base_in(180) <input type="radio"/> close_in(10) <input type="radio"/> go_to(34) <input type="radio"/> hold_in(15) <input type="radio"/> leave(34) <input type="radio"/> meet_in(22) <input type="radio"/> move_to(38) <input type="radio"/> open_in(28) <input type="radio"/> play_at(56) <input type="radio"/> report_from(11) <input type="radio"/> stop_in(13)	454/12
organization-organization :	<input type="radio"/> alternate_name(48) <input type="radio"/> approach(11) <input type="radio"/> beat(91) <input type="radio"/> buy(110) <input type="radio"/> compete(23) <input type="radio"/> cooperate(29) <input type="radio"/> create(40) <input type="radio"/> create_by(68) <input type="radio"/> deal_with(27) <input type="radio"/> join(17) <input type="radio"/> lose_to(33) <input type="radio"/> merge_with(26) <input type="radio"/> own(54) <input type="radio"/> own_by(71)	648/14
organization-person :	<input type="radio"/> accuse(30) <input type="radio"/> fire(21) <input type="radio"/> found_by(57) <input type="radio"/> head_by(83) <input type="radio"/> hire(57) <input type="radio"/> interview(23) <input type="radio"/> introduce(18) <input type="radio"/> lead_by(14) <input type="radio"/> lose(23) <input type="radio"/> praise(23) <input type="radio"/> promote(16) <input type="radio"/> sell(4) <input type="radio"/> send(33) <input type="radio"/> sign(64) <input type="radio"/> support(9)	475/15
person-location :	<input type="radio"/> bear_in(284) <input type="radio"/> campaign_in(23) <input type="radio"/> go_to(175) <input type="radio"/> grow_up_in(150) <input type="radio"/> leave(104) <input type="radio"/> like(19) <input type="radio"/> live_in(183) <input type="radio"/> represent(46) <input type="radio"/> rule(34) <input type="radio"/> speech_in(24) <input type="radio"/> study_in(13) <input type="radio"/> work_in(106)	1161/12
person-organization :	<input type="radio"/> appear_on(22) <input type="radio"/> call_on(39) <input type="radio"/> chairman_of(100) <input type="radio"/> create(97) <input type="radio"/> fire_by(19) <input type="radio"/> head(57) <input type="radio"/> help(39) <input type="radio"/> join(70) <input type="radio"/> leave(71) <input type="radio"/> member_of(54) <input type="radio"/> persuade(40) <input type="radio"/> study_at(47) <input type="radio"/> visit(45) <input type="radio"/> warn(21) <input type="radio"/> win(62)	783/15
person-person :	<input type="radio"/> accuse(140) <input type="radio"/> agree_with(15) <input type="radio"/> alternate_name(141) <input type="radio"/> father_be(20) <input type="radio"/> meet(174) <input type="radio"/> mother_be(22) <input type="radio"/> praise(71) <input type="radio"/> spouse_of(26) <input type="radio"/> telephone(94) <input type="radio"/> travel_with(23) <input type="radio"/> work_for(47) <input type="radio"/> work_with(126)	899/12
Total :	80 clusters and 4420 relations	

Relation Tag :

Relation id	<input type="checkbox"/> T1 <input type="checkbox"/> E1	<input type="checkbox"/> Cmid	<input type="checkbox"/> T2 <input type="checkbox"/> E2	Cpost	<input type="checkbox"/>
NYT_ENG_20050106.0038-16-1	golden cross farm	found by	allen	and his family be still	<input type="checkbox"/>
NYT_ENG_20060326.0118-9-1	carnegie hall	which be found by	andrew carnegie	scottish immigrant and once the	<input type="checkbox"/>
NYT_ENG_20041011.0330-5-1	kaiser aluminum	be found by progressive industrialist	henry j kaiser	who also found kaiser permanente	<input type="checkbox"/>
NYT_ENG_20041123.0157-42-1	shoah visual history foundation	establish by	spielberg	to record survivor ' memory	<input type="checkbox"/>
NYT_ENG_20060328.0334-4-2	bomb casualty commission	an agency establish by president	harry s truman	after world war ii to	<input type="checkbox"/>
Relation id	E1	Cmid	E2	Cpost	



Figure 3: Interface of our manual cluster construction tool

Type	reference	pre-filtering	post-filtering
ORG – ORG	454	307 (67.7%)	330 (72.7%)
ORG – LOC	648	485 (74.8%)	509 (78.5%)
ORG – PER	475	269 (56.6%)	286 (60.2%)
PER – ORG	1161	987 (85.0%)	998 (86.0%)
PER – LOC	783	597 (76.2%)	623 (79.6%)
PER – PER	899	586 (65.1%)	641 (71.3%)

Table 3: Global coverage of reference relations by clustering results

in reference clusters. Thus, classic measures such as *F-measure* can easily be defined to check how all $N(N-1)/2$ pairs of relations are grouped. A good clustering method should assign similar relations to the same cluster and dissimilar ones to different clusters. Hence, there are four kinds of decisions. First, a true positive (TP) decision assigns two similar relations to the same cluster while a true negative (TN) one assigns two dissimilar relations to different clusters. TP and TN are both correct decisions. On the other hand, there are two incorrect decisions: false positive (FP) decisions, which assign two dissimilar relations to the same cluster and false negative (FN) decisions, which assigns two similar relations to different clusters. Precision (P), recall (R) and F-measure are then classically defined as:

$$R = \frac{TP}{TP + FN} \quad P = \frac{TP}{TP + FP} \quad F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

Rather than examining all pairs of relations, clustering

quality can be evaluated directly at the cluster level with measures such as *Purity*, *Mutual Information* (MI) or *Normalized Mutual Information* (NMI). A pre-required step for computing such measures is to assign each final cluster to a reference cluster. The simplest strategy for performing such assignment is to choose the reference cluster that shares the largest number of relations with the considered final cluster. *Purity* is then defined by:

$$purity(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |w_k \cap c_j|$$

where $\Omega = \{w_1, w_2, \dots, w_K\}$ is the set of result clusters and $\mathbb{C} = \{c_1, c_2, \dots, c_J\}$ is the set of reference clusters. *Purity* has a bias when the number of clusters is large: it is equal to 1 when each relation forms its own cluster. Normalized mutual information makes a trade-off between the number of clusters and their quality. It is defined by:

$$NMI(\Omega, \mathbb{C}) = \frac{MI(\Omega, \mathbb{C})}{(H(\Omega) + H(\mathbb{C})) / 2}$$

where $MI(\Omega, \mathbb{C})$ is the mutual information between Ω and \mathbb{C} , with the definition:

$$MI(\Omega, \mathbb{C}) = \sum_k \sum_j P(w_k \cap c_j) \log \frac{P(w_k \cap c_j)}{P(w_k) * P(c_j)}$$

and $H(\Omega)$ and $H(\mathbb{C})$ are respectively entropies of Ω and \mathbb{C} , defined as:

$$H(\Omega) = - \sum_k P(w_k) \log P(w_k)$$

Type	Step	Precision	Recall	F-measure	TP	FP	FN	Purity	NMI
ORG-LOC	pre-filtering	0.977	0.246	0.393	5,029	120	15,416	0.694	0.648
	post-filtering	0.956	0.456	0.618	9,332	430	11,113	0.764	0.697
ORG-ORG	pre-filtering	0.984	0.309	0.471	6,264	100	13,982	0.789	0.756
	post-filtering	0.974	0.346	0.510	7,002	189	13,244	0.835	0.770
ORG-PER	pre-filtering	0.910	0.131	0.228	1,430	141	9,519	0.673	0.676
	post-filtering	0.932	0.152	0.262	1,668	122	9,281	0.753	0.689
PER-LOC	pre-filtering	0.676	0.409	0.510	39,525	18,981	57,009	0.770	0.627
	post-filtering	0.785	0.406	0.535	39,197	10,753	57,337	0.800	0.650
PER-ORG	pre-filtering	0.466	0.220	0.299	5,363	6,149	19,006	0.642	0.645
	post-filtering	0.395	0.274	0.323	6,667	10,192	17,702	0.618	0.621
PER-PER	pre-filtering	0.906	0.109	0.194	5,616	581	45,951	0.672	0.613
	post-filtering	0.875	0.120	0.211	6,181	883	45,386	0.738	0.604

Table 4: Evaluation with external measures (minority cases concerning the sense of the difference between pre and post-filtering values are *emphasized*)

Relation type	Relation	Clustering results
ORG – ORG	create	{create the}, {establish the}, {form a}, {build the}, ...
ORG – LOC	base in	{base in, a company base in}, {locate in, which be locate in}, {headquartered in}, ...
ORG – PER	found by	{found by, a group found by}, {be found by, which be found by}, {establish by}, ...
PER – ORG	head	{who head}, {who be the head of, who head the office of}, ...
PER – LOC	work in	{who work in}, {work in}, {work at}, {who work at}, ...
PER – PER	telephone	{call}, {who call, who call his manager}, {call president, telephone president}, ...

Table 5: Clustering results with filtering procedure

where $P(w_k)$, $P(c_j)$ and $P(w_k \cap c_j)$ are respectively the probabilities of a relation being in a result cluster w_k , in a reference cluster c_j and in the intersection of the two. The probabilities are computed directly by counting the cardinalities of the clusters.

The results of the application of these external measures to the clusters obtained by the method described in Section 2 with the reference described in Section 4.1 are shown in Table 4. We can first observe an improvement of the F-measure for all relation types when relations are filtered before they are clustered, which confirms our hypothesis that invalid relation instances have a negative influence on the clustering of relations. We can also note a satisfying level of precision both before and after relation filtering, especially for relation types such as ORG–ORG, ORG–LOC, ORG–PER and PER–PER. More precisely, the filtering of relations has globally a small negative impact on clustering precision but this impact is very limited for high precision values and only a little bit stronger for lower precision values, as for PER–LOC and PER–ORG relations. The decrease of precision due to relation filtering is globally compensated by the increase of recall, sometimes with a very significant difference as for ORG–LOC relations. The doubling of recall in this case results from the increase of TP decisions from 5,029 to 9,332, which clearly means that the presence of invalid relations can prevent a larger number of similar relations from being grouped together properly.

Purity and NMI measures are also improved in most of cases. However, it is difficult in principle to correlate directly these cluster level measures with F-measure for two main reasons. First, they can depend, as in the case of Pu-

riety, on the strategy that was chosen for assigning result clusters to reference clusters. Second, improvements of F-measure values tend to be more visible since this measure focuses on pairs of relations, whose number increases exponentially with the number of relations while the number of clusters increases more linearly with the number of relations. In practice, Table 4 shows that a positive impact of relation filtering on clustering is observed with all measures for all ORG–* and PER–LOC relations.

Finally, Table 5 provides a more qualitative view of relation clustering results by giving for each relation type (first column) one example of reference relation (second column) together with the clusters that were found by our clustering algorithm from the filtered relation instances of Section 2 and that can be associated to this reference relation. Each cluster is represented by the most frequent *Cmid* forms, appearing into curly brackets, among the various expressions covered by the cluster. A representative example of relation is given hereafter for each cluster associated with the relation *create* of type ORG–ORG shown in Table 5:

- *LAPD* creates the *Force Investigation Division* which probes potential criminal culpability ...
- *University of Florida* establishes the *Institute of Pharmacy Entrepreneur* last year to connect young ...
- *Stanford University* forms a *Global Climate & Energy Project* to combat global warming among ...
- *for the Kemper Development Company*, which is building a *Westin Hotel* topped by 148 condos...

A closer look at the content of clusters shows that a significant proportion of grouped relations share very similar expressions. This is not surprising as the measure we apply for evaluating the similarity between extracted relations is very basic but it has clearly a negative impact on clustering recall: a reference cluster is split among several built clusters, which prevents pairs of relations that are considered as similar in the reference from being identified as such in the clustering results. The most obvious way to improve this point is to define and to use a more elaborated similarity measure between extracted relations, in order to take into account semantic phenomena such as synonymy for instance and more globally to improve the detection of paraphrases. Strategies for grouping *a posteriori* clusters could also be considered and associated with the detection of similarity between relation instances for dealing with a wider range of variations among them.

5. Conclusion

In this article, we have tackled the problem of the evaluation of unsupervised information extraction. We have more particularly proposed two complementary ways to address it: a large-scale evaluation based on internal clustering measures that characterize to which extent clusters are representative of similarities between relations; a more restricted but deeper evaluation based on the *a priori* building of reference clusters and the use of external clustering measures. A methodology for building the reference clusters for a given corpus and an annotation tool for supporting this methodology have also been proposed by integrating a search engine and a simple ranking process. Finally, these evaluation methods have been applied to validate the interest of a filtering step in the unsupervised relation extraction process.

6. Acknowledgments

The research leading to these results has received funding from the European Union's Seventh Framework Program (FP7/2007-2013) under grant agreement n° SEC-GA-2009-242352 and from the French National Research Agency (ANR) FILTRAR-S project of the CSOSG 2008 program.

7. References

Michele Banko and Oren Etzioni. 2008. The Tradeoffs Between Open and Traditional Relation Extraction. In *ACL-08: HLT*, pages 28–36, Columbus, Ohio.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open Information Extraction from the Web. In *Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2670–2676, Hyderabad, India.

Roberto J. Bayardo, Yiming Ma, and Ramakrishnan Srikant. 2007. Scaling Up All Pairs Similarity Search. In *16th International Conference on World Wide Web*, pages 131–140, Banff, Alberta, Canada.

Oliviero Carugo. 2010. Clustering criteria and algorithms. In *Data Mining Techniques for the Life Sciences*, volume 609 of *Methods in Molecular Biology*, pages 175–196. Humana Press.

Edgar González and Jordi Turmo. 2009. Unsupervised relation extraction by massive clustering. In *Ninth IEEE International Conference on Data Mining (ICDM 2009)*, pages 782–787, Miami, Florida, USA.

María Halkidi, Yannis Batistakis, and Michalis Vazirgianis. 2002. Cluster validity methods: part I. *ACM SIGMOD Record (Special Interest Group on Management of Data)*, 31:40–45, June.

Julia Handl, Joshua Knowles, and Douglas B. Kell. 2005. Computational cluster validation in post-genomic data analysis. *Bioinformatics (Oxford, England)*, 21(15):3201–3212, August.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering Relations among Named Entities from Large Corpora. In *42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 415–422, Barcelona, Spain.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Benjamin Rosenfeld and Ronen Feldman. 2007. Clustering for unsupervised relation identification. In *Sixteenth ACM conference on Conference on information and knowledge management (CIKM'07)*, pages 411–418, Lisbon, Portugal.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive Information Extraction using Unrestricted Relation Discovery. In *HLT-NAACL 2006*, pages 304–311, New York City, USA.

Benno Stein, Sven, and Frank Wißbrock. 2003. On Cluster Validity and the Information Need of Users. In *3rd IASTED International Conference on Artificial Intelligence and Applications (AIA'03)*, pages 404–413.

Stijn van Dongen. 2000. *Graph Clustering by Flow Simulation*. Ph.D. thesis, University of Utrecht.

Wei Wang, Romaric Besançon, Olivier Ferret, and Brigitte Grau. 2011. Filtering and clustering relations for unsupervised information extraction in open domain. In *20th ACM Conference on Information and Knowledge Management (CIKM 2011)*, pages 1405–1414, Glasgow, Scotland, UK.