Filtering and clustering relations for unsupervised information extraction in open domain

W Wang, Romaric Besançon, Olivier Ferret, Brigitte Grau

Filtering and Clustering Relations for Unsupervised Information Extraction in Open Domain

Wei Wang[1]
wei.wang@cea.fr

Romaric Besançon[1]
romaric.besancon@cea.fr

Olivier Ferret[1]
olivier.ferret@cea.fr

Brigitte Grau[2]
brigitte.grau@limsi.fr

[1] CEA LIST, 18 route du Panorama, BP 6, Fontenay-aux-Roses, F-92265 France

[2] LIMSI, UPR-3251 CNRS-DR4, Bât. 508, BP 133, 91403 Orsay, France

ABSTRACT

Information Extraction has recently been extended to new areas by loosening the constraints on the strict definition of the extracted information and allowing to design more open information extraction systems. In this new domain of unsupervised information extraction, we focus on the task of extracting and characterizing *a priori* unknown relations between a given set of entity types. One of the challenges of this task is to deal with the large amount of candidate relations when extracting them from a large corpus.

We propose in this paper an approach for the filtering of such candidate relations based on heuristics and machine learning models. More precisely, we show that the best model for achieving this task is a Conditional Random Field model according to evaluations performed on a manually annotated corpus of about one thousand relations. We also tackle the problem of identifying semantically similar relations by clustering large sets of them. Such clustering is achieved by combining a classical clustering algorithm and a method for the efficient identification of highly similar relation pairs. Finally, we evaluate the impact of our filtering of relations on this semantic clustering with both internal measures and external measures. Results show that the filtering procedure doubles the recall of the clustering while keeping the same precision.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*clustering, information filtering, selection process*

General Terms

Algorithms, Experimentation

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

Keywords

Unsupervised information extraction, filtering, machine learning, clustering

1. INTRODUCTION

Traditionally, Information Extraction was considered from the viewpoint of the MUC (Message Understanding Conferences) paradigm [11]. According to this view, its objective is to extract pieces of information from texts for filling with a fixed role a predefined template. More recently, new forms of information extraction have been developed under the general idea of having more flexible ways to specify the information to extract from texts. Such information is generally defined as a configuration of relations between entities[1] with each relation being defined either as a handcrafted model (frequently a set of rules) or by a set of examples of relations in context that are used to train a statistical model. For an event such as an earthquake for instance, the extraction typically focuses on its location, its date, its magnitude and the damages it has caused and relies on the relations between these pieces of information and the mentions of the event [17]. This approach is globally a supervised or goal-driven approach. Weak forms of supervision have also been developed in this field. Work based on bootstrapping where, following [16], relations are first specified by a small number of examples or linguistic patterns [1], falls into this category. More recently, work related to the notion of *distant supervision* [23], in which relation examples are limited to pairs of entities without any linguistic form for relations, is also an example of such trend.

A reverse approach, called *unsupervised information extraction*, has also been explored during these last years. It aims at finding in texts relations between target entities or types of entities without any *a priori* knowledge concerning the type of the extracted relations. Furthermore, these relations can be clustered according to their similarity to be structured into meaningful sets. Work in this area can be considered according to three main viewpoints. The first one regards the unsupervised extraction of relations as a means for learning knowledge. This view has been developed both for learning "general world knowledge" through the concept of *Open Information Extraction* [2] applied for large-scale knowledge acquisition from the Web in [3] and in more re-

[1]Configuration that is often restricted to one relation.

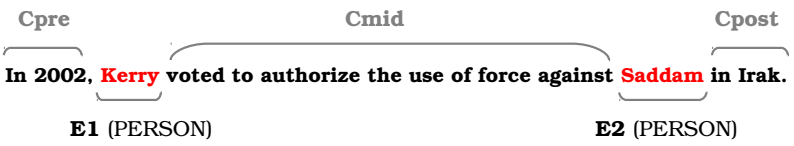

Figure 1: Example of extracted relation

stricted domains, as the biomedical domain, where such relation extraction is used for adding new types of relations between entities in an already existing ontology [6].

The two other viewpoints are more directly related to Information Extraction. The main one tackles the problem of making it possible for users to specify their information needs in a more open and flexible way. The *On-demand information extraction* approach [26], relying on [14] and extended by the notion of *Preemptive Information Extraction* [27], aims at inducing a kind of *template* from a set of documents that are typically retrieved by a search engine from queries that are representative of the information to extract. The same perspective can be found in [19] and, with a specific emphasis put on relation clustering, in [25].

Finally, the last viewpoint, less represented than the two others, considers unsupervised information extraction as a source of improvement for supervised information extraction. The supervised approach frequently depends on manually annotated corpora. As the task is complex, the annotation cost is high and these corpora are generally not very large. In this context, the results of an unsupervised approach can be used to extend the coverage of models learned from an annotated corpus. This idea is more specifically developed in [4] and is also present in [10].

Following the second viewpoint above, our work contributes to the definition of a more flexible information extraction scheme. Within this context, it tackles more particularly the problem of the filtering of extracted relations. The objective of such information extraction process is to discover new types of relations between entities by clustering relations, which requires as less noise as possible among the extracted relations. Hence, we first focus on the filtering procedure, whose objective is to determine whether a relation exists between two named entities in a sentence without any *a priori* knowledge about its type. Then, we evaluate the impact of such filtering on the clustering of large sets of relations and show that it leads to double its recall with the same precision.

2. OVERVIEW

The work we present in this article takes place in a larger context whose global objective is to develop an unsupervised information extraction process for addressing technology watch issues such as "tracking all events involving companies X and Y". This process is based on the extraction of relations defined initially by the co-occurrence of two named entities in a sentence, similarly to most of the works cited in the previous section[2]. The main idea behind these restrictions is to focus first on simple cases to counterbalance the difficulties raised by the unsupervised nature of the global approach. "Simple cases" means here relations whose arguments are rather easy to identify and relations whose linguistic expression is small enough to be easily delimited and to avoid coreference phenomena concerning their arguments.

More formally, candidate relations extracted from texts are characterized by two different kinds of information:

- a pair of named entities (E1 and E2);
- the linguistic form of the relation. It refers more precisely to the way the relation is expressed. As relation extraction is based on the presence of two named entities in a sentence, the linguistic form is made of three parts of this sentence:
  - *Cpre*: part before the first entity (E1);
  - *Cmid*: part between the two entities;
  - *Cpost*: part after the second entity (E2).

  The core expression of the relation is generally conveyed by *Cmid* while *Cpre* and *Cpost* are more likely to bring context elements that are used for detecting its similarity with other relations in the perspective of their clustering.

Figure 1 gives an example of relation with its constituents. It should be noted that such relation has a semi-structured form as one part of its definition – the pair of entities – is defined with elements coming from an already existing ontology while its other part only appears under a linguistic form.

The unsupervised information extraction process based on this notion of relation is defined by the following sequence of tasks:

- analysis of documents (linguistic preprocessing);
- extraction of candidate relations;
- filtering of extracted candidate relations;
- clustering of relations according to their similarity.

The linguistic preprocessing of documents aims at extracting the defining elements of relations. Hence, it includes named entity recognition for the target types of entities but also part-of-speech tagging and lemmatization for normalizing the three parts of the linguistic description of relations. It is achieved by the OpenNLP tools[3].

The step of candidate relation extraction involves very limited constraints: all pairs of named entities with the target types are extracted provided that the two entities appear in the same sentence with at least one verb between them. Table 1 illustrates the volume of the extracted candidate relations from a subpart of the AQUAINT-2 corpus containing news articles from 18 months of the newspaper *New*

[2]In some works such as [2], entities in relation are extended to any noun phrases.

[3]`http://opennlp.sourceforge.net`

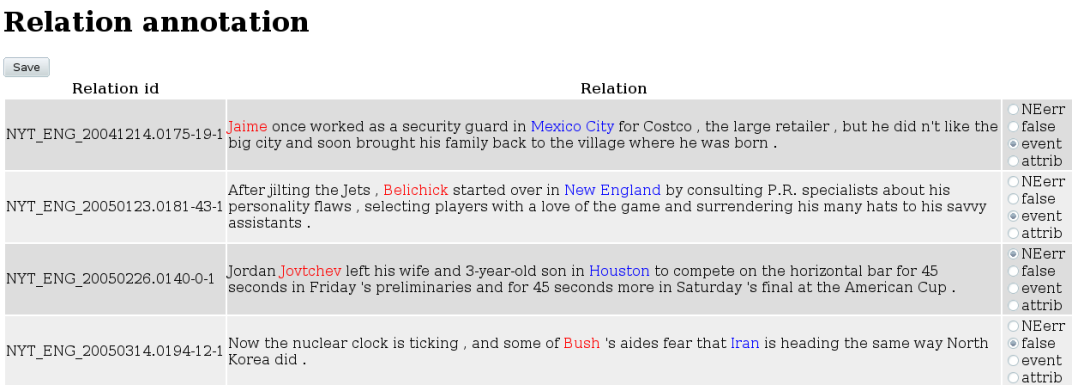

Figure 2: Interface for relation annotation

York Times. All our experiments in this article about relation filtering in section 3 and relation clustering in section 4 are based on this corpus and focus on relations involving persons (PER), organizations (ORG) or locations (LOC).

Table 1: Volume of extracted relations

Relation type	Number of relations
LOC – LOC	116,092
LOC – ORG	57,092
LOC – PER	78,845
ORG – LOC	71,858
ORG – ORG	77,025
ORG – PER	73,895
PER – LOC	152,514
PER – ORG	126,281
PER – PER	175,802

3. RELATION FILTERING

As a consequence of our relation extraction strategy, a significant number of candidates do not contain a true relation between their entities. This basic method, which can lead to good results in specific domains ([9] shows that 79% of extracted candidates using this heuristic are true relations in the biomedical domain), does not seem to be selective enough for open domain. Therefore, we added to this simple strategy a filtering procedure designed to determine the existence of a relation between two entities in a sentence.

3.1 Filtering Heuristics

We first defined more discriminative criteria to filter out sentences that do not contain a true relation between a pair of entities. In this perspective, three heuristics were tested:

- elimination of relations that contain a discourse related verb between the two entities (the list of verbs is currently limited to *to say* and *to present*). This aims at avoiding to extract a relation between the entities *Homgren* and *Allen* in a sentence like: "*Holmgren* said *Allen* was more involved with the team ...";
- the maximum distance between the two entities is limited to 10 words. Effective relations become very rare beyond this empirical limit;
- only one verb is allowed between the two entities, except auxiliary verbs (*be*, *have* and *do*): we discard sentences with a too complex syntactic structure between the entities as they tend to make the existence of a relation between them less likely.

The application of these three heuristics to relation extraction globally reduced the volume of candidate relations by about 50%. Table 2 presents in more details the filtering ratio of each relation type for a sample of 8,000 relations for each type. For each pair of entity types, the second column shows the numbers of relations filtered and kept using all the heuristics, as well as the ratio of kept relations. The three following columns give the number of filtered relations for each individual heuristic, considering that one relation may be filtered by more than one heuristic. The distance limit has obviously an important filtering effect but the only-one-verb limitation has an equally significant impact.

These filtering ratios only give quantitative information about the reduction of relation candidates. In order to evaluate the efficiency of this heuristic filtering in a reliable and feasible way, a subset of 50 randomly selected relations of each type were manually annotated to verify their validity with a Web interface (presented in Figure 2) generated by applying XSLT transformations to the XML representation of relations.

Table 3: Evaluation of filtering heuristics

Relation type	Filtered		Kept	
	true	false	true	false
LOC – LOC	1	49 (98%)	9 (18%)	41
LOC – ORG	4	46 (92%)	8 (16%)	42
LOC – PER	3	47 (94%)	2 (4%)	48
ORG – LOC	7	43 (86%)	14 (28%)	36
ORG – ORG	6	44 (88%)	20 (40%)	30
ORG – PER	4	46 (92%)	20 (40%)	30
PER – LOC	13	37 (74%)	40 (80%)	10
PER – ORG	12	38 (76%)	40 (80%)	10
PER – PER	5	45 (90%)	14 (28%)	36

The results of this annotation, presented in Table 3, show that a very high percentage of filtered relations are indeed false ones, which confirms the relevance of our filtering criteria. These results also show that the ratios of false relations

Table 2: Effect of the application of filtering heuristics on a sample of 8,000 relations

Relation type	filtered/kept	discourse	distance	one verb
LOC − LOC	4287/3713 (46%)	440	3548	2763
LOC − ORG	4097/3903 (49%)	488	3224	2650
LOC − PER	4790/3210 (40%)	1636	3352	2638
ORG − LOC	4225/3775 (47%)	643	3324	2869
ORG − ORG	4169/3831 (48%)	627	3123	2810
ORG − PER	4541/3459 (43%)	1541	3155	2859
PER − LOC	4209/3791 (47%)	905	3199	2813
PER − ORG	3888/4112 (51%)	952	2742	2566
PER − PER	4444/3556 (44%)	1290	3109	2741

after filtering remain important, especially those with LOC as first named entity type. This phenomenon can be explained by the fact that, in a true relation, the first entity should have an agent role in the sentence whereas location names often occur at the beginning of sentences in adverbial phrases. These cases could be detected using a deeper syntactic analysis but such analysis is too costly for the amount of data we process. Considering this observation, relations between entities involving a LOC as first entity are excluded from the following steps.

3.2 Filtering by Machine Learning

The results of Table 3 demonstrate the utility of our filtering heuristics. However, it also indicates that these heuristics are not sufficient to reach a high proportion of correct relations (*i.e.* high enough for the following steps of the unsupervised information extraction process). We present in this section an additional filtering method using statistical machine learning models. Training and test corpora for these models were built manually by annotating relations with the same interface as in Figure 2. More precisely, 200 relations for each of the 6 pairs of our target entity types were randomly selected and annotated. The annotation distinguished correct relations (*true*), incorrect relations due to a named entity recognition error (*NEerr*), and incorrect relations due to the absence of any effective relation (*false*). An additional distinction between the nature of the relations was also made (between attributive relations and event-related relations) but it was not exploited in this work. The results of this annotation are presented in Table 4.

Table 4: Results of manual annotation

Relation Type	true	NE errors	false
ORG − LOC	38% (77)	18% (35)	44% (88)
ORG − ORG	39% (78)	14% (28)	47% (94)
ORG − PER	36% (72)	18% (36)	46% (92)
PER − LOC	51% (102)	31% (62)	18% (36)
PER − ORG	60% (120)	18% (36)	22% (44)
PER − PER	41% (82)	20% (39)	40% (79)
All	44% (531)	20% (236)	36% (433)

The figures of Table 4 show that about 20% incorrect relations are caused by named entity errors. These relations were removed from the corpus to avoid introducing too much noise in the training data. The remaining corpus was composed of 964 relations, 531 of which were true and 433 were false. The resulting set of relation instances is well-balanced enough to avoid problems related to the training of classifiers with unbalanced data sets.

3.2.1 Non-Structured Local Feature Models

We first tested several models based on non-structured local features. Classically, we trained a Naive Bayes classifier, a Maximum Entropy classifier (MaxEnt), a Decision Tree and a Support Vector Machine classifier (SVM). The first three models were implemented using the tools provided by MALLET [22] while the last model was implemented with SVM^{light} [18].

The same set of features was used to train these four different classifiers:

- type of named entities E1 and E2;
- Part-of-Speech (POS) of words between the two entities, using a binary feature for each pair $< P_i, POS_i >$ (with P_i, the position of current word in *Cmid*), as well as bigrams of POS between the two entities, using a binary feature for each triplet $< P_i, POS_i, POS_{i+1} >$;
- POS for the two words before E1 and the two words after E2, both with unigrams and bigrams;
- POS sequence for words between E1 and E2: each possible sequence of 10 POS was encoded as a binary feature;
- number of tokens between E1 and E2;
- number of punctuation marks (comma, quotation mark, parenthesis ...) between E1 and E2.

3.2.2 Sequential Model for Machine Learning Filtering

As [4], we also tested a classifier based on the sequential tagging of each word in a sentence. Our representation of this sequential model is illustrated by Figure 3.

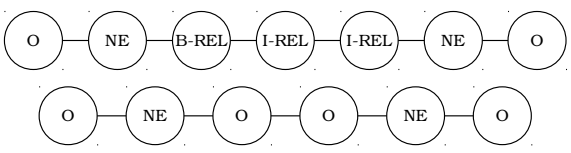

Figure 3: Sequential representation of sentence annotation

More precisely, each word is tagged with one of the four labels below, following the BIO encoding model introduced by [24]:

Table 5: Evaluation of statistical classifiers

Model	Accuracy	Precision	Recall	F_1-measure
Naive Bayes	0.637	0.660	0.705	0.682
MaxEnt	0.650	0.665	0.735	0.698
Decision Tree	0.639	0.640	0.784	0.705
SVM	0.732	0.740	0.798	0.767
CRF	0.745	0.762	0.782	0.771
[4]	/	0.883	0.452	0.598

- O: words not related to a relation or named entity;
- NE: named entity that defines a potential relation (E1 or E2);
- B-REL: first word after E1 inside a relation;
- I-REL: continuation of a relation after B-REL.

After their tagging, sentences containing true relations are labeled as the first configuration of Figure 3 (with a variable number of I-REL depending on the expression of the relation) whereas false relation sentences are labeled as the second one. In practice, a well-trained classifier should not generate configurations other than the two presented in Figure 3: for instance (O – NE – B-REL – O – O – NE – O) is not possible since B-REL is always followed by at least one I-REL in the training corpus.

This approach was implemented by a linear Conditional Random Fields (CRF) model, using the Wapiti [20] tool, and trained with the following set of features for each word of the sentence:

- POS of the current word, the previous one and the following one;
- bigrams of POS $<\text{POS}_{i-1}, \text{POS}_i>$, with i=-1,0,1 (0: current word; -1: previous word; 1: following word);
- entity type of the current word and the 6 previous and following words. This type is equal to NULL when the word is not a named entity.

3.2.3 Evaluation of Statistical Filtering

Considering the relatively small size of the annotated corpus, we used a 10-fold cross-validation to evaluate these different classifiers: the corpus was split into 10 equal parts, 9 of which being used for training and 1 for testing; the procedure is repeated 10 times so that each part is used for training and for test at least once. The results of Table 5 represent the average values on the 10 iterations of the standard measures of *Accuracy*, *Precision*, *Recall* and *F1-measure*.

Table 5 first shows that the SVM classifier obtains the best performance among the non-sequential models, which is a result generally obtained by similar works about relation extraction. It also shows that the CRF classifier slightly outperforms the SVM classifier, which confirms the advantage of the sequential model. Moreover, we note a balance between precision and recall for all types of classifiers. Compared with the results of [4] on the same subject (see the last line of the table), our results exhibit a better F1-measure, with a much better recall and a slightly lower precision. However, the work in [4] relies on more general entities than only named entities, which makes the task harder. As we have not enough room here for detailing the various relation representations and the different sets of features we have tested, we only present those achieving the best performance. However, it is interesting to notice, concerning the entities of relations, that removing the entity type as a feature and replacing it by a generic "NE" tag to mark the presence of a named entity only causes a slight decrease of the performance of the CRF classifier, with a F1-measure equal to 0.768. This indicates a promising extensibility of our classifier to other named entity types. Finally, as a consequence of the global results of this evaluation, the CRF model was adopted for the relation filtering part of our unsupervised information extraction method in the experiments of section 4.

3.3 Application of Relation Filtering

The extraction of relations is composed of three successive steps:

- initial extraction, only based on the the co-occurrence of two named entities with the target types and the presence of at least one verb in between;
- application of three filtering heuristics for eliminating a large number of false relations with a good precision;
- application of a machine learning filtering to distinguish more finely true relations from false ones.

Moreover, we observed the presence of a certain number of identical relations coming either from articles about the same subject or from very formatted expressions. Hence, we completed the filtering procedure with a final deduplication step for discarding these redundant relations. As there exists a superior boundary for the values of the similarity measure between relations, the implementation of this final step was based on the identification of pairs of relations with this maximal similarity value, which was done by relying on the same approach as for the clustering of relations in section 4 for evaluating the similarities between relations. For relations having this maximal similarity value, only one representative element was kept. We put this deduplication operation as the last step of the relation extraction process for two reasons: first, this procedure is more costly than the other filtering operations; second, it relies on the evaluation of relation similarity performed for relation clustering.

Table 6 shows detailed information for each step of relation filtering, starting from all candidate relations of Table 1. We can note that this filtering put aside a large number of the initially extracted relations but we have estimated that only 19.9% of these discarded relations result from erroneous decisions, with a global recall of the filtering procedure estimated to 0.553. Finally, the remaining volume is *a priori* sufficient for the next steps of our unsupervised information

Table 6: Relation volumes after each filtering step

	ORG-LOC	ORG-ORG	ORG-PER	PER-LOC	PER-ORG	PER-PER
initial extraction	71,858	77,025	73,895	152,514	126,281	175,802
heuristics	33,505 (47%)	37,061 (48%)	32,033 (43%)	72,221 (47%)	66,035 (52%)	78,530 (45%)
classifier CRF	16,700 (23%)	17,025 (22%)	12,098 (16%)	55,174 (36%)	50,487 (40%)	42,463 (24%)
deduplication	15,226 (21%)	13,704 (18%)	10,054 (14%)	47,700 (31%)	40,238 (32%)	38,786 (22%)

extraction process. Furthermore, as in [4], the context of our work is the processing of large text collections characterized by informational redundancy for which high-precision results are preferred to avoid too much noise.

4. RELATION CLUSTERING

4.1 Method

The objective of our work is to cluster similar relations in order to offer users a better view of existing relations between entities as in many articles in the domain of unsupervised information extraction [27, 25]. We have chosen for this clustering an approach similar to [14]: we only consider one level of clustering for gathering relations that share the same meaning (semantic clustering). The definition of this similarity of meaning is relatively loose: it does not only represent a strict notion of paraphrase or implication but is more related to the notion of information redundancy used in automated summarization.

The clustering method relies on two elements: a similarity measure between relations and a clustering algorithm that uses the pairwise similarities between relations. For the similarity measure, we chose the widely used *cosine* measure and applied it to a bag-of-words representation of relations. More precisely in our case, we only used the *Cmid* part of each relation in order to focus on its core meaning rather than on its context. The choice of the cosine measure was also justified by the results of preliminary experiments showing its superiority over the edit distance for a similar task.

Clustering algorithms often rely on a similarity matrix which can be costly to compute, in particular for larges sets of relations like the one we want to process (several tens of thousands of relations), since the number of similarities is quadratic with respect to the number of relations. Algorithms such as *k-means* are a little less costly since they only consider the similarities between the points and the centroids of current clusters but they require to fixed *a priori* the number of classes, which is hard to evaluate in our case (optimizing this number is possible but can lead again to a problem of complexity). We tackled this issue by using the *All Pairs Similarity Search* algorithm (APSS) [5] that allows to compute efficiently a similarity measure such as the cosine measure for all pairs of elements whose similarity value is above a given threshold. The efficiency of this algorithm relies on a series of optimizations in the indexing of the elements to compare that exploit the fixed threshold and the sparsity of the input vectors for reducing the number of comparisons to perform. In our experiments, this threshold was based on observations from the Microsoft Research Paraphrase Corpus [8]. This corpus contains a set of sentence pairs associated with an assessment indicating if they are paraphrases or not. We computed the cosine measure for all pairs of paraphrase sentences and chose to fix our threshold value to 0.45, which covers 3/4 of the similarity values between these sentences.

We then used the *Markov Clustering* algorithm [29] to create the final clusters of relations from the similarity matrix computed by the APSS algorithm. More precisely, this matrix, which is rather sparse, is directly transformed into a similarity graph by associating each relation with a node and each non-zero similarity with a weighted edge between two nodes. The Markov Clustering algorithm performs the partitioning of a graph by the means of a series of random walks on the graph. This algorithm converges quite fast in practice, which allows to deal with large graphs, and does not depend on a fixed number of clusters: its only parameter, *inflation*, controls the granularity of the clusters. In our experiments, we adopted the default inflation value of the MCL implementation[4].

4.2 Evaluation of Relation Clustering

We present in this section a quantitative evaluation of the relation clustering. Evaluation of clustering is a hard task because no gold-standard partitioning of the set of elements is available: such a reference would be too costly to build, considering we have tens of thousands of relations. The usual approach in this case is to evaluate the quality of the results obtained by manually looking at a particular set of clustering results. A major drawback of this kind of evaluation is that it relies on a specific clustering configuration and would require to perform the complete evaluation process again if the clustering technique changes. We wanted to have a more reproducible evaluation framework in order to compare clustering results with and without filtering and possibly with different filtering techniques.

We then propose to perform two evaluations of the clustering: the first one is an evaluation using internal criteria; the second one using external criteria, but only on a partial reference. On one hand, internal criteria for clustering evaluation allows to establish to which extent the clusters obtained correspond to the similarity measures between the relations [12]. More precisely, we use the internal criteria to test the hypothesis that the similarities in the relation space after filtering have a better distribution that the ones before filtering, then leading to a better clustering. On the other hand, external criteria allow to better take into account an actual evaluation of whether two relations in the same cluster belong to the same semantic relation. Since we do not have the possibility to create a gold-standard for the whole set of relations, we decided to create reference data for a selected subset of relations and evaluate how these relations are distributed among the different clusters.

4.2.1 Clustering Evaluation with Internal Measures

Among various internal measures for clustering evalua-

[4]http://micans.org/mcl

tion, we chose a measure of *expected density*, which is evaluated in [28] as the one having the best correlation with F-measure for documents clustering (the more usual measure of the *Dunn index* is said to be less stable).

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 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 is defined by:

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

where $\frac{|V_i|}{|V|}$ intends to balance the difference of size of clusters. For taking into account the considerable difference of the collection size due to the filtering phase, we defined an expected density measure that is less dependent on the corpus size by loosening the exponential factor $|V_i|$, which is connected to the size of each cluster. Therefore, we used the following definition of expected density:

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

A higher value of the measure ρ' implies a better clustering quality.

We also considered the *Connectivity* measure [13], another internal measure. Connectivity evaluates how many nearest neighbors are not clustered together. This measure is of particular interest for us since it is based on the same similarity graph that we are using for the clustering method. The connectivity measure is defined by:

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

where p denotes how many neighbors are taken into account, $nn_i(j)$ 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.

As shown by its formal definition, connectivity also depends on corpus size. To avoid such dependence, we selected randomly a subset of the total corpus (5,000 relations were used for evaluation in our experiments). This measure is inverse compared to the expected density: a lower connectivity value indicates a better clustering.

Results of expected density measure and connectivity measure are presented in Table 7. The results with these two internal measures show that the filtering phase generally improves the clustering processing. Better clusters are generated from the filtered relations using the same clustering method. The two entity pairs which 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. Actually, as we stated in section 3, location entities are often included in adverbial phrases. When such a case happens, there is no real relation between the location entity and the other entity although, with the currently used similarity measure, phrases with similar location adverbials can be clustered together and obtain a good clustering score.

4.2.2 Clustering Evaluation with External Measures

The first results with internal measures demonstrate the interest of the filtering procedure. Then, we have tried to confirm this interest using external measures by comparing the clustering results with reference clusters. A partial reference has been built in three steps:

1. indexing of all relation candidates with a search engine;
2. querying of this index iteratively to locate interesting relations;
3. creation of relation clusters manually from the results of the queries.

More precisely, we first indexed the extracted relations with the search engine Lucene[5], using distinct fields for the text, the named entities and the entity types. This allows us to search relations with queries specifically targeting their first or second named entity (*E1*, *E2*), the types of these entities (*T1*, *T2*) or the linguistic constituents of relations *Cmid*, *Cpos* or *Cpre* (see relation example in Figure 1). We used this possibility by querying the index with various combinations of fields, in a first step to explore potential target relations between given named entities and entity types (*e.g.* with queries such as *E1=Bush, T2=*LOC), and then in a second step, to explore different named entities with target relations (*e.g.* with queries such as *T1=*PER, *Cmid contains "visit"*). After several iterations of these two steps, we obtained a set of relations mixing a large diversity of relations together with a significant number of similar relations. Based on these relations, we built manual clusters with a specific Web-based annotation tool.

Currently, our gold-standard reference concentrates on the relation type PER – LOC and contains 17 clusters with 253 relations, including relations such as, *come from*, *be going to*, *have a speech in*, *like*, etc. We present below some examples for the relation *grow up in*:

- *Pitcher Brandon Backe*, who grew up 50 miles from here in *Galveston* and dreamed of pitching for the Astros in the postseason, displayed a veteran's savvy with his varying speeds.
- Chief Justice Wallace B. Jefferson, a Republican, named *Pat Priest*, a retired Democratic judge from his hometown of *San Antonio*, to hear the case – but not before Jefferson's own multiple ties to DeLay's political operation were questioned.
- By the time he turned 10, *Levine* had performed as a soloist with his hometown *Cincinnati* Orchestra.

External measures like *Purity*, *Normalized Mutual Information* and *Rand Index* are well discussed in the literature (*e.g.* [21]). Given reference clusters with N relations, *Rand Index* is defined to check how all $N(N-1)/2$ pairs of relations are grouped. A clustering method should assign similar

[5]http://lucene.apache.org

Table 7: Internal evaluation for relation clustering (best results are presented in bold)

	Expected density		*Connectivity* ($p = 20$)	
	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 8: External evaluation for relation type PER − LOC

phase	*rand index*	*precision*	*recall*	F_1
pre-filtering	0.8863	0.5372	0.1271	0.2056
post-filtering	0.8864	0.5280	0.2211	0.3117

TP	*FP*	*FN*	*TN*
469	404	3221	27784
866	774	3051	28979

purity	*MI*	*NMI*
0.5540	1.6338	0.6008
0.5889	1.7216	0.6285

relations to the same cluster and separate dissimilar ones. 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. The *Rand Index* measures the clustering accuracy, which is defined by:

$$RandIndex = \frac{TP + TN}{TP + FP + FN + TN}$$

F-measure can be defined in the same time, relying on the precision P and recall R:

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN} \quad F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

We have computed the distribution of reference relations in clustering results for both the pre-filtering phase and the post-filtering phase. The results are presented in Table 8. These results show that the filtering procedure almost doubles the recall measure, from 0.1271 to 0.2211, while the precision is kept around 0.53. We can also see directly from this table that many more pairs of truly similar relations (TP) are found by the clustering method on the post-filtering corpus than on the pre-filtering corpus.

Rather than examining all pairs of relations, clustering quality can be measured directly at a cluster level. Purity and *Normalized Mutual Information* (NMI) are usually used for this purpose. A prerequisite of this approach is to assign each result cluster to the class (a reference cluster) with which it shares the largest number of relations. *Purity* is defined by:

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

where $\Omega = \{w_1, w_2, ...w_K\}$ is the set of result clusters and $\mathbb{C} = \{c_1, c_2, ..., c_J\}$ 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:

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

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

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

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

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

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.

In the same way as for the *Rand Index* measure, we computed these measures for the result clusters obtained using relations extracted with or without the filtering phase. Evaluation results for *Purity* and *NMI* are illustrated in Table 8. The results show that both *Purity* and *Normalized Mutual Information* are improved by the filtering. In particular, the augmentation of *Purity* confirms the recall improvement observed with the *Rand Index* measure.

5. RELATED WORK

One of the specificities of the work we have presented in this article is to associate two types of work in the field of unsupervised information extraction, relation filtering and relation clustering, and to study the consequences of this association. Concerning relation filtering, our work is close to the work described in [4], with two main differences. First, relation arguments are restricted to named entities in our case, whereas such argument has more general form in [4] and can be any base noun phrase[6]. Second, [4] does not rely on manually annotated examples as we do but exploits examples built automatically from the successful parses of a

[6]Base noun phrases refer to noun phrases that do not contain nested noun phrases or modifiers such as prepositional phrases.

syntactic parser by applying a small set of heuristics. The possible impact of these two differences is not easy to predict as they tend to diverge in terms of effects. The first difference makes relation extraction more difficult in the case of [4] as it enlarges the set of possible relations to cover. However, restricting the possible relations to a subset of the successful parses of a syntactic parser clearly favors relations with a simple syntactic form whereas the limits of our reference relations are only set by a human annotation. Finally, the results of [4], a high precision but a low recall, can be explained as follows: because of the kind of entities it focuses on, [4] considers a large set of possible relations but the classifier it has developed is actually able to takes into account only a small subset of them because of the way it is trained. Our approach implements a more balanced choice between the set of relations we want to take into account and the set of relations we actually model, which globally leads to higher results. From a practical viewpoint, building the training examples automatically from the results of a syntactic parser as [4] did is of course an interesting choice to have a large training set without the cost of a human annotation but of course, this method heavily depends on both the availability and the quality of such parser in the target context (language, domain or type of texts).

Concerning relation clustering, the comparison with existing work raises two main issues. The first one is the scalability of the clustering process. Clustering algorithms frequently start from a similarity matrix that can be difficult to compute when the number of items to cluster is large. One way to overcome this difficult is to fix or to evaluate *a priori* the number of clusters to build. For instance, [30] sets arbitrarily the number of clusters according to the document set, [25] tests different values whereas [10] uses the Akaike Information Criterion to evaluate this number in one of its experiments. This problem is also bypassed in some works by limiting the number of relations to cluster, either directly or through the initial number of documents. [14] for instance only considers relations with at least 30 occurrences whereas experiments in [25] are limited to 4,000 relation occurrences and those in [30] to 526 Wikipedia documents. In our case, this issue is tackled by associating a filtering method for discarding explicitly false relations and the use of the APSS algorithm for evaluating efficiently the similarity of the remaining relations. This combination makes the use of a large spectrum of clustering algorithms possible.

The second important issue concerning relation clustering is the evaluation of its results. As mentioned in section 4.2, a direct evaluation of the built clusters and their content by human annotators as it was performed in [14] or in [30] cannot be achieved very often because of its cost. In particular, it does not fit the constraints resulting from the tuning of a system. It is why we have adapted and applied measures for the internal evaluation of clustering to the context of unsupervised information extraction, which was not done before to our knowledge. These measures were more specifically used for testing the impact of relation filtering and their conclusions appear as coherent with those of external measures, as illustrated in section 4.2.2. For our external evaluation, we have chosen as [25] to select a sample of relations and to cluster them manually to build a reference. More precisely, because of the very large number of relations we have, this selection was guided in our case by a search engine. Finally, the evaluation consists in determining to what extent relations that are part of a reference cluster are found in the same cluster in the evaluated clustering. Following [15], [10] adopts the same principle but uses as reference the relations annotated in a corpus in the context of a supervised information extraction task, more precisely, the Relation Mention Detection task of the ACE (Automatic Content Extraction) evaluation [7].

6. CONCLUSION AND PERSPECTIVES

In this paper, we have presented a work on relation filtering for unsupervised information extraction whose purpose is to determine if two entities occurring in the same sentence are linked by a relation without *a priori* knowledge about the relation type. This filtering is performed using both heuristic and machine learning techniques. Heuristic filtering is first used to remove the simple cases whereas machine learning techniques are used for more ambiguous cases. Evaluation of machine learning techniques shows that best results are obtained with CRF, compared with results obtained with SVM, MaxEnt or NaiveBayes classifiers. Our best performances are quite balanced between precision and recall and are better than the results reported in [4] (but their study is not limited to named entities, which makes the task more difficult). Applied to unsupervised information extraction, we have also showed, through an evaluation of relation clustering with both internal and external criteria, that this filtering is useful for a semantic clustering of the extracted relations.

The most direct perspectives of this work are about the clustering of relations. We will improve the external evaluation of the clustering using a larger set of annotated examples and try to have more evidences on the interest of such partial external evaluation by computing its correlation with a completely annotated reference. We also plan to use a more sophisticated clustering including two levels of clustering: a semantic clustering and a thematic clustering. Finally, we also consider applying this filtering process to improve a system for knowledge base population based on *distant supervision* by filtering out the candidate relations extracted from a corpus for learning linguistic relation patterns.

7. ACKNOWLEDGMENTS

This work was partly supported by the ANR FILTRAR-S project and the FP7 Virtuoso project.

8. REFERENCES

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In 5^{th} *ACM International Conference on Digital Libraries*, pages 85–94, San Antonio, Texas, USA, 2000.

[2] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open Information Extraction from the Web. In *Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2670–2676, Hyderabad, India, 2007.

[3] M. Banko and O. Etzioni. Strategies for Lifelong Knowledge Extraction from the Web. In 4^{th} *International Conference on Knowledge Capture (K-CAP 2007)*, pages 95–102, Whistler, BC, Canada, 2007.

[4] M. Banko and O. Etzioni. The Tradeoffs Between Open and Traditional Relation Extraction. In *48th Annual Meeting of the ACL: Human Language Technologies (ACL-08: HLT)*, pages 28–36, Columbus, Ohio, 2008.

[5] R. J. Bayardo, Y. Ma, and R. Srikant. Scaling Up All Pairs Similarity Search. In *16th International Conference on World Wide Web*, pages 131–140, Banff, Alberta, Canada, 2007.

[6] M. Ciaramita, A. Gangemi, E. Ratsch, J. Saric, and I. Rojas. Unsupervised Learning of Semantic Relations between Concepts of a Molecular. Biology Ontology. In *International Joint Conference on Artificial Intelligence (IJCAI 2005)*, pages 659–664, Edinburgh, UK, 2005.

[7] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The Automatic Content Extraction (ACE) Program – Tasks, Data, and Evaluation. In *4th Conference on Language Resources and Evaluation (LREC 2004)*, pages 837–840, 2004.

[8] B. Dolan, C. Quirk, and C. Brockett. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *20th International conference on Computational Linguistics (COLING 2004)*, pages 350–356, Geneva, Switzerland, 2004.

[9] M. Embarek and O. Ferret. Learning patterns for building resources about semantic relations in the medical domain. In *6th Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, 2008.

[10] E. González and J. Turmo. Unsupervised Relation Extraction by Massive Clustering. In *Ninth IEEE International Conference on Data Mining (ICDM 2009)*, pages 782–787, Miami, Florida, USA, 2009.

[11] R. Grishman and B. Sundheim. Design of the MUC6 evaluation. In *MUC-6 (Message Understanding Conferences)*, Columbia, MD, 1995. Morgan Kauffmann Publisher.

[12] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Cluster validity methods: part I. *ACM SIGMOD Record (Special Interest Group on Management of Data)*, 31:40–45, June 2002.

[13] J. Handl, J. Knowles, and D. B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics (Oxford, England)*, 21(15):3201–3212, August 2005.

[14] T. Hasegawa, S. Sekine, and R. Grishman. Discovering Relations among Named Entities from Large Corpora. In *42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 415–422, Barcelona, Spain, 2004.

[15] H. Hassan, A. Hassan, and O. Emam. Unsupervised Information Extraction Approach Using Graph Mutual Reinforcement. In *2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*, pages 501–508, Sydney, Australia, 2006.

[16] M. Hearst. Automatic Aquisition of Hyponyms from Large Text Corpora. In *14th International Conference on Computational Linguistics (COLING'92)*, pages 539–545, Nantes, France, 1992.

[17] L. Jean-Louis, R. Besançon, and O. Ferret. Using Temporal Cues for Segmenting Texts into Events. In *7th International Conference on Natural Language Processing (IceTAL 2010)*, pages 150–161, Reykjavik, Iceland, 2010.

[18] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press, Cambridge, MA, 1999.

[19] H. H. Kathrin Eichler and G. Neumann. Unsupervised Relation Extraction From Web Documents. In *6th Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, 2008.

[20] T. Lavergne, O. Cappé, and F. Yvon. Practical Very Large Scale CRFs. In *48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 504–513, Uppsala, Sweden, 2010.

[21] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[22] A. K. McCallum. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu, 2002.

[23] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore, 2009.

[24] L. Ramshaw and M. Marcus. Text Chunking Using Transformation-Based Learning. In D. Yarowsky and K. Church, editors, *Third Workshop on Very Large Corpora*, pages 82–94, Cambridge, Massachusetts, USA, 1995.

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

[26] S. Sekine. On-Demand Information Extraction. In *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 731–738, Sydney, Australia, 2006.

[27] Y. Shinyama and S. Sekine. Preemptive Information Extraction using Unrestricted Relation Discovery. In *HLT-NAACL 2006*, pages 304–311, New York City, USA, 2006.

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

[29] S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[30] Y. Yan, N. Okazaki, Y. Matsuo, Z. Yang, and M. Ishizuka. Unsupervised Relation Extraction by Mining Wikipedia Texts Using Information from the Web. In *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2009)*, pages 1021–1029, Suntec, Singapore, 2009.