# Semantic clustering of relations between named entities

W. Wang, R. Besançon, Olivier Ferret, B. Grau

# Semantic Clustering of Relations between Named Entities

Wei Wang[1], Romaric Besançon[1], Olivier Ferret[1], and Brigitte Grau[2]

[1] CEA, LIST, Vision and Content Engineering Laboratory
91191 Gif-sur-Yvette Cedex, France
wei.wang@lip6.fr,romaric.besancon@cea.fr,olivier.ferret@cea.fr
[2] LIMSI, UPR-3251 CNRS-DR4, Bat. 508,
BP 133, 91403 Orsay Cedex, France
brigitte.grau@limsi.fr

**Abstract.** Most research in Information Extraction concentrates on the extraction of relations from texts but less work has been done about their organization after their extraction. We present in this article a multi-level clustering method to group semantically equivalent relations: a first step groups relation instances with similar expressions to form clusters with high precision; a second step groups these initial clusters into larger semantic clusters using more complex semantic similarities. Experiments demonstrate that our multi-level clustering not only improves the scalability of the method but also improves clustering results by exploiting redundancy in each initial cluster.

**Keywords:** unsupervised information extraction, relation extraction, clustering

## 1 Introduction

Unsupervised Information Extraction (UIE) differs from standard Information Extraction (IE) approaches by opening IE systems to unknown information structures. Such approaches allow to discover non-predefined relations between entities [12], which helps handling the heterogeneous relation types found in open-domain [2, 9] and proves useful in application contexts such as strategic or competitive intelligence. A very light form of supervision can also be taken into account in such approaches by enabling users to delimit a topical context, such as in the *On-Demand Information Extraction* paradigm [23].

Most of the work in UIE is dedicated to the extraction of relations and less to their organization. Based on a statistical classifier (*e.g.*, TextRunner [2]), on bootstrapping (*e.g.*, Ollie [16]) or on patterns and rules (*e.g.* ReVerb [9] or [1, 11]), such systems concentrate on guaranteeing the validity of each extraction rather than on the organization of relations. However, structuring the extracted relations is important both to characterize the type of relations and facilitate the access to information for the end-users.

In [14], a clustering of relations is performed but misses a more semantic dimension in relation similarity and fails to group synonyms or paraphrases. A more semantic similarity is proposed by [7] but is evaluated only on a small corpus. Large-scale semantic clustering of extracted relations is still a challenge, even if some approaches have been proposed such as [18] or [19]. However, [18] applies semantic criteria mainly for finding equivalent entities, whereas we focus on similar relation expressions, and [19] exploits the specific context of Wikipedia.

We present in this article an efficient and effective multi-level clustering procedure that succeeds in grouping relations that are expressed either by similar expressions or synonymous phrases and can be applied at a large scale. We focus only on relations between named entities because, in an applicative context of strategic or competitive intelligence, relations of interest are mostly oriented by named entities. Experiments demonstrate that our multi-level clustering not only improves the scalability of the method, but also improves the clustering results by exploiting redundancy in each initial cluster.

## 2 Related Work

In UIE approaches based on clustering [12, 22, 23], clustering methods play a dual role since they provide a cluster structure to relation instances at the same time these instances are extracted. In [12], each cluster is likely to contain semantic variations of the same relation, including synonyms, since it results from the merging of sets of relation instances based on the co-occurrence of the same pair of named entities. [23] creates an off-line base of paraphrases, relying on shared named entities to align sentences from multiple newspapers reporting the same event: relation patterns linking the same pair of entities are placed in the same pattern set. In general, these clustering methods are designed to detect reliable relation patterns while we are interested in a method that focuses specifically on finding synonymous patterns.

[14] proposes a method for extracting high-level relations and concepts from the relations of TextRunner through a co-clustering method based on Markov Logic that simultaneously generates classes of arguments and classes of relations. However, as it doesn't exploit any lexical semantic resources, it generally fails to group synonyms or paraphrases. In the same way, [4] performs co-clustering but applies it on a dual representation of relations, either as entity pairs (extension) or as lexical patterns (intension). This co-clustering relies on a matrix of co-occurrence between entity pairs and lexical patterns and requires, to be effective, a good connectivity in the entity relation graph. Similarly, the effectiveness of the inference procedure of generative models in UIE [21, 27] relies on the connectivity of the whole entity relation graph, which is not very scalable. Our multi-level clustering approach addresses this issue by limiting the use of semantic similarity measures to small sets of relation instances. [18] also clusters both the relations and their arguments but adopts a less integrated approach and relies on lexical semantic resources built from corpora. Its clustering method globally takes advantage of redundant information from a first clustering so that

more equivalent entities can be detected. However, our objective is more focused on the detection of equivalent expressions of relations.

Concerning semantic similarities, [7] exploits lexical information from WordNet but its evaluation is done on a small corpus whereas we target large-scale approaches. Moreover, it only exploits verbs, whereas we want to include nouns as well, and we rely on an initial clustering step that provides a more robust base to our semantic clustering.

## 3 A Multi-level Clustering

We propose a multi-level clustering procedure for relation organization that groups the instances of relations extracted from a large corpus into clusters by relying on their semantic similarity.

Each relation instance is extracted from the co-occurrence of two named entities in a sentence and then filtered according to the two-step method defined in [24]: a filtering based on simple heuristics (such as a threshold on the number of words between the entities) is first applied to throw away a large number of incorrect relations with a good precision, followed by a second, more fine-grained, filtering based on a statistical classifier. Relation instances are then characterized by a pair of named entities and a linguistic form, composed of the normalized part of the sentence between these entities, called *Cmid*.

In unsupervised IE tasks, the number of extracted relation instances can be very large, which makes the direct search for semantic similarities among these instances too costly. At the same time, we need to deal with the diversity of these instances both in terms of types and forms of expression. We also observed that this diversity can be decomposed into several levels and more particularly, that a part of the extracted relation instances can be described by the same keyword, with slight variations, as illustrated in Table 1. In this table, each line refers to the same relation, expressed with different linguistic forms.

**Table 1.** Examples of variations of the linguistic form of relations

| Category | Relations, grouped by form, with normalized words | |
|---|---|---|
| ORG – ORG | create the, who create, ... | establish the, who establish the, ... |
| ORG – LOC | base in, a company base in, ... | locate in, which be locate in, ... |
| ORG – PER | a group found by, which be found by, ... | |
| PER – ORG | who be the head of, become head of, ... | |

The large-scale constraint, the variability of the expressions of the relations and this observation motivate our multi-level approach: by first clustering relations with very similar linguistic expressions (such as *create the* and *who create*), we can efficiently group the syntactic paraphrases of a relation into small but precise initial clusters. Then, a second level of clustering takes into account more complex semantic similarities to further group the initial clusters into larger semantic clusters. The cost of using semantic criteria is limited by the fact that the initial clusters are far less numerous than the extracted relations. Both initial clustering and semantic clustering are applied within each relation category, characterized by the type of the named entities linked by the relation.

### 3.1 Initial Clustering

As stated above, the goal of the initial clustering is to split the large set of extracted relations into groups of similar relations with only slight syntactic differences. To implement this kind of similarity between relations, we used the standard *Cosine* similarity on a bag-of-word representation of the *Cmid* part of the relation, which is an interesting option due to its efficiency. Moreover, this calculation was made faster by the application of the *All Pairs Similarity Search* (APSS) algorithm [3], which builds the similarity matrix of relations very efficiently by exploiting a fixed similarity threshold, given as a parameter, to avoid the computation of all pairwise similarities.

All the words in the linguistic expression of a relation do not have the same importance, which we characterize in our framework by a weight. More precisely, we experimented three methods for weighting the words of relations.

**Binary** All words appearing in *Cmid* are given the same weight (w=1.0).

**tf-idf** Words are weighted by the *tf-idf* score. *idf* corresponds in our case to the inverse relation frequency and measures the specificity of a word among the extracted relations.

**POS** Specific weights are given to words according to their part-of-speech (POS) category. An analysis of POS categories led us to divide them into four classes (plus a default class with weight w=0.5) according to the importance of their contribution to the semantic expression of a relation:

*Direct* (w=1.0) words directly linked to the meaning of a relation, including verbs, nouns, adjectives, prepositions and particles;

*Indirect* (w=0.75) words that are not directly linked to the meaning of the relation but are characteristic of a form of its expression, such as adverbs and pronouns;

*Complement* (w=0.5) words that provide complementary information in the relation, such as proper nouns and interjections;

*Noise* (w=0.0) words, such as symbols, numbers, determiners, coordinating conjunctions or modal words, that are not considered as relevant to the expression of the relation.

This initial clustering procedure groups most similar expressions into the same cluster in a precise way. Nevertheless, some relation instances are missed because the general weighting schemes do not always give high weights to the most significant words of the linguistic form of a relation. Furthermore, we observed that most of the relation instances are characterized by either a verb (e.g.

*founded* for *a group founded by*, *which is founded by*) or a noun (e.g. *head* for *who is the head of*, *becomes head of*). In an initial cluster, this characterizing keyword has generally a much higher frequency than the other words. Hence, following [12], we consider the most frequent word (verb or noun) of an initial cluster as its *label* and use it to add a refinement step to the initial clustering in which the initial clusters that share the same label are merged to form bigger initial clusters.

### 3.2 Semantic Clustering

The second clustering step aims at grouping the initial clusters according to a more semantic similarity in order to gather equivalent relations expressed differently, such as *based in* and *which is located in*. This clustering relies on cluster-to-cluster comparisons, which actually implies to consider three levels of semantic similarity. The first level is the targeted similarity between the initial clusters to merge. This similarity relies on the similarity between relations, which are the basic elements of clusters, which itself relies on a semantic similarity between words because they are the basic elements of the linguistic form of the relations. We describe how these three levels of semantic similarity are implemented in the increasing order of their granularity (words, relations, clusters).

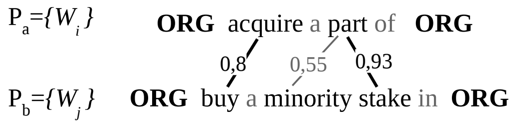
**Word-level similarity.** Semantic similarity measures between words are usually separated into two categories: measures based on manually-built resources such as WordNet and distributional measures, based on corpus-based data. We compared the two types of measures for our task of semantic clustering.

*WordNet-based measures.* Numerous types of measures were proposed to compute similarities between the synsets of WordNet by relying on their hierarchy. We considered two measures that are complementary and representative of different families of measures: on the one hand, the measure from Wu and Palmer [26] ($Sim_{wup}$), which takes into account both the depth of the two synsets to compare in the WordNet hierarchy and the depth of their least common subsumer; on the other hand, the measure from Lin [15] ($Sim_{lin}$), which also includes statistical information about synsets (Information Content) derived from a corpus.

These similarities are defined between synsets, each of which may contain several words. In the same way, each word may be included in different synsets. A simple way of mapping synset similarity to word similarity is to choose the highest synset similarity among all possible synset pairs [17].

*Distributional measures.* Distributional measures are based on the distributional hypothesis that words occurring in the same context tend to have similar meanings. Practically, given a large corpus, a set of co-occurrents, either extracted from a fixed size window or from syntactic dependency relations, is collected for each word to form its *context vector* and the semantic similarity of two words is evaluated by computing a standard bag-of-word similarity measure between their contexts, such as *Cosine*, *Jaccard* or *Dice* [13].

**Relation-level similarity.** The evaluation of the semantic similarity of two relations is related to the problem of paraphrase recognition. More precisely, as each relation is represented by its *Cmid* part, the considered problem can be seen as the evaluation of the similarity of two phrases $P_a$ and $P_b$, represented as bag-of-words. A simple way of computing the similarity of two phrases or sentences is to take the average of the word-level similarities between all possible word pairs. However, all word pairings do not have the same relevance, especially for two words that are not important in the expression of the relation. [17] proposed to match each word in one phrase only with the most similar word in the other phrase and to only take into account these most similar matches, as illustrated by the following example, where *part* is only paired with *stake*:



The similarity is then given by the following equation:

$$S_1(P_a, P_b) = \frac{1}{\sum_{W_i \in P_a} w_i} \sum_{W_i \in P_a} \max_{W_j \in P_b} \{S_{W_{i,j}}\} \cdot w_i \tag{1}$$

where $w_i$ is the weight given to the word $W_i$ in $P_a$ and $S_{W_{i,j}}$ is the similarity of words $W_i$ and $W_j$ computed following the various options of word-level similarity.

With this definition, the measure is not symmetric $(S_1(P_a, P_b) \neq S_1(P_b, P_a))$: $W_i$ in $P_a$ being the most similar word with $W_j$ in $P_b$ does not guarantee that $W_j$ is the most similar word in $P_b$ for $W_i$. Therefore, the average of similarities in both directions is taken to make this measure symmetric, defined by:

$$S(P_a, P_b) = \frac{1}{2}\big(S_1(P_a, P_b) + S_1(P_b, P_a)\big) \tag{2}$$

**Cluster-level similarity.** Each initial cluster contains two or more relation instances. A complete-linkage or average-linkage between clusters is very costly since it requires to compute the similarities between all relation pairs from the clusters. On the other hand, choosing only one relation instance randomly as a representative of an initial cluster is not a reliable procedure, even with the high precision of each cluster. Moreover, the definition of an average linguistic representation for a cluster is not always obvious and may result in an important loss of information, especially when this information was collected without known expectations.

The proposed solution is to merge the bag-of-word representations of all the relation instances of an initial cluster to form a general bag-of-words for this initial cluster $C = \{W_i{:}f_i\}$, where each word is associated with its frequency in the cluster. The hypothesis is that the most relevant words with respect to the relation appear more frequently in the cluster and should be given a higher weight. The frequency of words in initial clusters is considered as representative of the information redundancy in these clusters. Therefore, the same similarity as the relation-level similarity (Equation 1) can be adopted. However, this weighting scheme faces a frequency bias problem. Let two clusters $C_a = \{\text{found:3}, \text{actor:3}\}$ (*an actor who found*) and $C_b = \{\text{study:9}, \text{actor:1}\}$ (*study at*, *an actor who study at*), which are not semantically similar. However, the similarity from $C_a$ to $C_b$ is high because the shared word *actor* has a high frequency in the first cluster. Even though the inverse similarity (from $C_b$ to $C_a$) is low, the average similarity is strongly influenced by the first one and has a relatively high level. To solve this frequency bias problem, the frequencies of matching words in both clusters are taken into account for the computation of similarity in each direction. This leads to replace, in Equation 1, the weight $w_i$ by the weight $w_{ij}$, defined as: $w_{ij} = f_i \cdot f_j$.

**The choice of clustering algorithms** The performance of clustering algorithms largely depends on the nature of the considered data and the specific constraints of the targeted tasks. In unsupervised IE tasks, the clustering algorithms must have the capacity to process large data sets and to deal with the unpredictable number of clusters (due to the heterogeneity of open-domain relations). Hierarchical clustering algorithms are often too costly for large data sets and other standard methods such as K-means require a predefined number of clusters. In this study, we considered Markov Clustering (MCL) [6] and Shared Nearest Neighbors clustering (SNN) [8], which are both efficient and do not require a predefined number of clusters. The MCL algorithm generally requires a pruning threshold to ignore all unnecessary values in the similarity matrix for efficiency and noise filtering (which can also be an advantage from a computational point of view since it can be combined with APSS, presented in section 3.1). The SNN algorithm can be very efficient, even without a pruning threshold on similarity, but it is highly parameterized, and most importantly, the number of nearest neighbors to consider is not obvious to determine in all cases. MCL was chosen for the initial clustering step because specifying a similarity threshold corresponds intuitively to fix the proportion of common words between two phrases, whereas the sizes of clusters can be very diverse in an open-domain context. On the contrary, SNN was chosen for the semantic clustering step because the number of neighbors to consider, which is a central parameter of this method, refers to some extent to the average number of synonymous words or paraphrases, which is more stable than the values of the semantic similarity[3].

## 4 Experiments and Evaluations

### 4.1 Evaluation Measures and Dataset

For the evaluation of all our clustering results, we used measures both at the level of relations and clusters. At the relation level, the *precision* (*prec.*) and

[3] These choices have also been verified practically by experiments: SNN results are much worse than MCL results for initial clustering and better for semantic clustering.

*recall* measures were applied on pairs of relation instances, considering that the relations can be positively or negatively grouped into the same cluster or separated in different clusters. At the cluster level, the evaluation was performed with the standard *purity*, *inverse purity* (*inv. purity*) and *Normalized Mutual Information* (NMI) measures. Our experiments were performed on the 159,400 documents of the *New York Times* part of the AQUAINT-2 corpus but the evaluation focused on a set of 4,420 relations extracted from this corpus and manually grouped into 80 clusters [25]. These relations are divided into the six categories of Table 4 according to the types of the named entities in relations. It is important to note that, unlike other evaluations such as in [18] or [19], these reference clusters were built *a priori* and are not the result of the judgment of automatically built clusters. Hence, they represent a less biased form of reference than the usual ones in the field.

### 4.2 Initial Clustering Experiments

The similarity threshold used to prune the matrix similarity (using the APSS algorithm) was set to 0.45 for the binary weighting MCL (*i.e.* the association of MCL with a binary weighting of words in relation instances). This threshold was set empirically for covering 3/4 of the similarity values between similar sentences based on observations from the Microsoft Research Paraphrase Corpus [5]. The validity of this threshold was confirmed in practice since it outperforms other tested values, ranging from 0.35 to 0.60. The same threshold was adopted for the tf-idf weighting MCL algorithm whereas, considering the looser constraints in the weighting with POS categorization, a more strict threshold was used (0.60) in this case. The results of the initial clustering using the different weighting strategies are presented in Table 2.

**Table 2.** Results of the initial clustering with different word weighting strategies

| | **Prec.** | **Recall** | **F-score** | **Purity** | **Inv. purity** | **NMI** | **#clusters** | **Size** |
|---|---|---|---|---|---|---|---|---|
| **binary** | 0.756 | 0.312 | 0.442 | 0.788 | 0.407 | 0.671 | 15,833 | 7.50 |
| **tf-idf** | 0.203 | 0.445 | 0.279 | 0.646 | 0.573 | 0.712 | 11,911 | 11.44 |
| **POS** | **0.810** | 0.402 | **0.537** | **0.867** | 0.513 | **0.739** | 13,648 | 7.56 |
| **Refinement** | **0.812** | **0.443** | **0.573** | 0.857 | **0.552** | **0.751** | 11,726 | 8.80 |

The MCL algorithm with the similarity measure based on POS weighting outperforms the other two weighting configurations, with a better precision and a relatively satisfying recall. This is understandable since this weighting strategy takes more knowledge into account to emphasize the importance of verbs, nouns, adjectives and prepositions, which carry the meaning of the relation, and gives less weight to words that contribute mainly to linguistic variations ("who" + verb, "the one that" + verb). This distinction enables the pruning threshold to be increased to improve precision without any big loss of recall.

On the other hand, the tf-idf weighting does not lead to good results. It tends to favor words that are rather rare in the corpus while verbs and nouns that are meaningful to relations are often rather frequent and thus, have a small weight. For instance, the verb "*buy*" is frequent in financial documents, which leads to a low *idf* value, but holds the key role in relation instances for the relation BUY(ORG,ORG). On the contrary, words such as proper nouns or specific numbers, which are not linked with the relation type, are not frequent and often obtain a higher weight. As a result, the td-idf weighting disturbs the clustering of relations by producing irrelevant similarities.

As a consequence, the results of the initial clustering kept for the semantic clustering step were the results obtained with the POS weighting strategy[4], on which the refinement procedure was applied. Table 2 shows that this refinement step leads to a slight improvement of F-score, especially due to the increase of recall; but it is also important to note that this step succeeds in reducing the number of clusters and increasing their average size, as illustrated in the last two columns of Table 2.

### 4.3 Semantic Clustering Experiments

We evaluated the different methods for semantic clustering presented in section 3.2 and compared them with a theoretical upper-bound result for this step (that we call "*ideal*" clustering), defined to be the best possible performance for semantic clustering, given the results of the initial clustering: each initial cluster is associated with the reference cluster that shares the largest number of relation instances with it; then, the initial clusters that are associated with the same reference cluster are grouped to form the new *ideal semantic clusters*[5].

**Evaluation of Semantic Similarities.** The semantic clustering was applied on the initial clusters resulting from the POS weighting initial clustering with refinement. As stated previously, the important words characterizing a relation are generally verbs and nouns. For our semantic similarity, we have chosen to compare only words with the same part-of-speech, with the objective of grouping relation instances that are either mainly characterized by verbs, such as *found by* or *establish by*, or mainly characterized by nouns, such as *be partner of* or *have cooperation with.*

In practice, for WordNet-based measures, the $Sim_{wup}$ similarity performs well for noun-noun comparisons while the $Sim_{lin}$ similarity performs better for verb-verb comparisons. For nouns, we used the *Wup* similarity as implemented by the NLTK package (`nltk.org`) while for the *Lin* similarity, we exploited the pre-computed similarity pairs from [20].

[4] Several different pruning thresholds and different weighting configuration for POS categories have been tested. The presented version (threshold 0.60 and POS weighting) is the one that gives the best results.

[5] Note that this ideal clustering is therefore performed only on the initial clusters that contain relations present in the 80 reference clusters.

For the distributional similarities, we used the distributional thesaurus built by [10], obtained using co-occurrences in a window of size 3 (which means only one nearest content word was considered on each side) on the whole AQUAINT-2 corpus. We also tested a distributional similarity based on co-occurrences obtained from syntactic relations. For both types of distributional similarities, only the three most similar words were taken into account. For the SNN clustering algorithm, we limited the number of considered neighbors to 100. The results of the semantic clustering based on these different similarity measures are presented in Table 3.

**Table 3.** Results of semantic clustering with different similarity measures, compared to the upper-bound *Ideal* clustering

| | **Prec.** | **Recall** | **F-score** | **Purity** | **Inv. purity** | **NMI** | **#clusters** | **Size** |
|---|---|---|---|---|---|---|---|---|
| WordNet | 0.821 | 0.507 | 0.627 | 0.846 | 0.622 | 0.763 | 9,403 | 10.98 |
| Window-based | 0.814 | 0.540 | 0.649 | 0.836 | 0.634 | 0.764 | 10,161 | 10.16 |
| Syntax-based | **0.831** | **0.549** | **0.661** | **0.853** | **0.645** | **0.770** | 10,116 | 10.20 |
| Ideal | 0.861 | 0.701 | 0.773 | 0.867 | 0.770 | 0.797 | 13,468 | 7.66 |

The syntax-based distributional similarity achieves the best performance but is comparable to the performance achieved by the window-based distributional similarity. Both distributional similarities outperform WordNet-based similarities, which means that this method can be quite easily adapted to other languages since distributional resources are much easier to obtain than manually built lexical resources such as WordNet. Compared to initial clustering results, all semantic similarities succeed in improving both precision and recall.

Concerning the choice of the words on which the similarity is applied, we observed[6] that the performance using only verbs for semantic clustering is a bit inferior to the results using both nouns and verbs. Taking nouns into account especially improves the recall measure and increases the average cluster size. The integration of adjectives in the similarity computation was also experimented but showed a very limited influence on the final results. Moreover, we tested cross-category distributional similarities between verbs and nouns, with no obvious improvement in recall or precision.

**Semantic clusters examples.** Table 4 gives a qualitative view of the semantic clustering results by presenting some examples of semantic clusters formed using the syntax-based distributional similarity. Each word corresponds to the label of an initial cluster. These examples show that distinct words that are semantically similar, and even distant paraphrase forms such as *grab gold in* and *win the race at*, are grouped together. However, certain errors are still present, for instance the presence of *purchase* and *be purchased by* in the same cluster due to the lack of differentiation between passive and active forms by our current preprocessing.

[6] We do not present all the quantitative results due to lack of space.

**Table 4.** Semantic clustering results

| Category | Semantic clusters |
|---|---|
| ORG – ORG | purchase, buy, acquire, trade, own, be purchased by |
| ORG – LOC | start in, inaugurate service to, open in, initiate flights to |
| ORG – PER | sign, hire, employ, interview, rehire, receive, affiliate |
| PER – ORG | take over, take control of |
| PER – LOC | grab gold in, win the race at, reign |
| PER – PER | win over, defeat, beat, topple, defend |

### 4.4 The Effects of Multi-Level Clustering

As discussed in section 3.1, computing semantic similarities is much more time-consuming than simple *Cosine* similarities. The total number of relation instances reaches up to 165,708 while the number of initial clusters is only 11,726. Therefore, a first advantage of our multi-level clustering is to avoid the heavy calculation of semantic similarities on a huge set of relation instances.

A second advantage is that it exploits the redundancy of information in initial clusters to identify interesting elements and to improve the quality of the semantic clustering. To validate this hypothesis, we compared, based on our reference, the distribution of similarities between relation instances and the distribution of similarities between initial clusters. First, we examined all the similarities between two relation instances in the same reference cluster (intra-distribution $D_{intra}$) and all the similarities between two instances in different reference clusters (inter-distribution $D_{inter}$). Ideally, the two distributions $D_{intra}$ and $D_{inter}$ should be well separated, with a high average similarity for $D_{intra}$ and a low average similarity for $D_{inter}$. Secondly, we associated each reference cluster with the set of initial clusters it covers[7] and we examined all the similarities between two initial clusters in the same reference cluster, which forms a new intra-distribution $D'_{intra}$, and all the similarities between two initial clusters in different reference clusters, which forms a new inter-distribution $D'_{inter}$.

These distributions are presented in Figure 1 for the syntax-based distributional similarity (but similar results are obtained with all types of semantic similarity), with $D_{intra}$, $D_{inter}$ on the left, and $D'_{intra}$, $D'_{inter}$ on the right. It is clear from these graphs that the semantic clustering based on the initial clusters achieves a more stable performance since intra and inter-distributions are better separated and initial clusters in different reference clusters have a rather low similarity on average. This confirms our hypothesis that redundant information in initial clusters can be used to filter out the noise brought by irrelevant words.

## 5 Conclusion and Perspectives

We present in this paper a multi-level approach for clustering relation instances extracted in the framework of unsupervised information extraction. This method

[7] Since our initial clustering method tends to form small but precise clusters, each reference cluster is split into several small clusters.

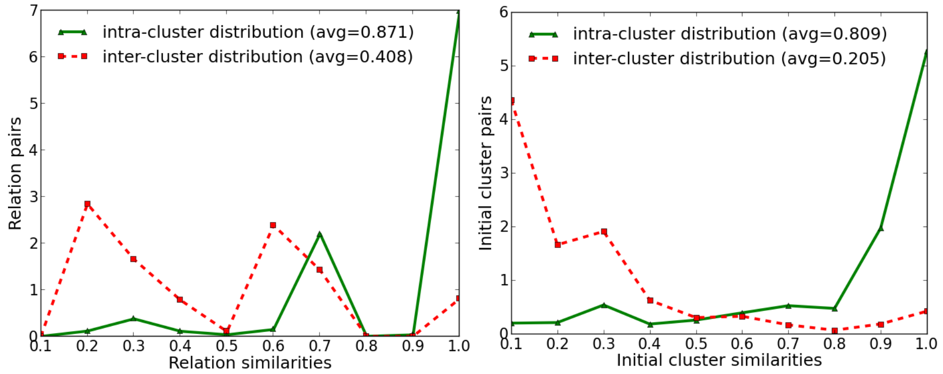


**Fig. 1.** Distribution of similarities between relations and between initial clusters

deals with both problems of scalability and linguistic diversity of relations by using two levels of clustering: a first level builds small clusters in an efficient and precise way while a second level, more semantic, relies on different semantic similarities between words including WordNet-based and distributional similarities. We demonstrate in our experiments the interest of this approach.

In future work, we consider expanding the semantic similarities to take more information into account and using deeper syntactic information about the linguistic expression of the relations in order to spot more precisely the interesting elements of the relations. We also started some experiments to combine the semantic clustering of the relations to a thematic clustering of the contexts in which they appear to give a more precise background for the relation definition.

**Acknowledgments.** This work was partly funded by European Union Seventh Framework Program FP7/2007 - 2013 under grant agreement n°FP7-SEC-2012-312651.

## References

1. Akbik, A., Broß, J.: Extracting semantic relations from natural language text using dependency grammar patterns. In: SemSearch 2009 workshop (2009)
2. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: IJCAI'07. pp. 2670–2676 (2007)
3. Bayardo, R.J., Ma, Y., Srikant, R.: Scaling up all pairs similarity search. In: WWW'07. pp. 131–140 (2007)
4. Bollegala, D.T., Matsuo, Y., Ishizuka, M.: Relational Duality: Unsupervised Extraction of. Semantic Relations between Entities on the Web. In: WWW'10. pp. 151–160 (2010)
5. Dolan, B., Quirk, C., Brockett, C.: Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In: COLING'04. pp. 350–356 (2004)

6. Dongen, S.V.: Graph Clustering by Flow Simulation. Ph.D. thesis, University of Utrecht (2000)
7. Eichler, K., Hemsen, H., Neumann, G.: Unsupervised relation extraction from web documents. In: LREC'08. pp. 1674–1679 (2008)
8. Ertöz, L., Steinbach, M., Kumar, V.: A New Shared Nearest Neighbor Clustering Algorithm and its Applications. In: Workshop on Clustering High Dimensional Data and its Applications of SIAM ICDM 2002. pp. 105–115 (2002)
9. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: EMNLP'11. pp. 1535–1545 (2011)
10. Ferret, O.: Testing Semantic Similarity Measures for Extracting Synonyms from a Corpus. In: LREC'10. pp. 3338–3343 (2010)
11. Gamallo, P., Garcia, M., Fernández-Lanza, S.: Dependency-Based Open Information Extraction. In: Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP. pp. 10–18 (2012)
12. Hasegawa, T., Sekine, S., Grishman, R.: Discovering relations among named entities from large corpora. In: ACL'04. pp. 415–422 (2004)
13. Heylen, K., Peirsmany, Y., Geeraerts, D., Speelman, D.: Modelling Word Similarity: An Evaluation of Automatic Synonymy Extraction Algorithms. In: LREC 2008. pp. 3243–3249 (2008)
14. Kok, S., Domingos, P.: Extracting Semantic Networks from Text Via Relational Clustering. In: ECML PKDD'08. pp. 624–639 (2008)
15. Lin, D.: An Information-Theoretic Definition of Similarity. In: ICML'98. pp. 296–304 (1998)
16. Mausam, Schmitz, M., Soderland, S., Bart, R., Etzioni, O.: Open language learning for information extraction. In: EMNLP-CoNLL 2012. pp. 523–534 (2012)
17. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI'06. pp. 775–780 (2006)
18. Min, B., Shi, S., Grishman, R., Lin, C.Y.: Ensemble Semantics for Large-scale Unsupervised Relation Extraction. In: EMNLP'12. pp. 1027–1037 (2012)
19. Moro, A., Navigli, R.: Integrating syntactic and semantic analysis into the open information extraction paradigm. In: IJCAI 2013. pp. 2148–2154 (2013)
20. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet::Similarity - Measuring the Relatedness of Concepts. In: HLT-NAACL 2004: Demonstrations. pp. 38–41 (2004)
21. Rink, B., Harabagiu, S.: A generative model for unsupervised discovery of relations and argument classes from clinical texts. In: EMNLP'11. pp. 519–528 (2011)
22. Rozenfeld, B., Feldman, R.: High-Performance Unsupervised Relation Extraction from Large Corpora. In: ICDM'06. pp. 1032–1037 (2006)
23. Sekine, S.: On-Demand Information Extraction. In: COLING-ACL'06. pp. 731–738 (2006)
24. Wang, W., Besançon, R., Ferret, O., Grau, B.: Filtering and Clustering Relations for Unsupervised Information Extraction in Open Domain. In: CIKM 2011. pp. 1405–1414 (2011)
25. Wang, W., Besançon, R., Ferret, O., Grau, B.: Evaluation of unsupervised information extraction. In: LREC'12. pp. 552–558 (2012)
26. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: ACL'94. pp. 133–138 (1994)
27. Yao, L., Haghighi, A., Riedel, S., McCallum, A.: Structured relation discovery using generative models. In: EMNLP'11. pp. 1456–1466 (2011)