Learning Selectional Preferences for Entailment or Paraphrasing Rules

Basili Roberto, Diego De Cao, Paolo Marocco, Marco Pennacchiotti
Department of Computer Science, System and Production
University of Roma, *Tor Vergata*
Via del Politecnico, 1 - 00133 Roma (Italy)
{*basili,decao,marocco,pennacchiotti*}@*info.uniroma2.it*

Abstract

Recent work about textual entailment or paraphrasing emphasizes the role of automatic learning of inference rules. Major weakness of these repositories is the low accuracy reachable in applying the rules in operational settings (e.g. textual entailment challenges or question answering). In this paper a robust method for automatic learning of inference rules is presented. As opposed to existing proposals, it relies on a geometrical model of similarity, based on a form of latent semantic analysis applied to the source text collection. The result is a non merely distributional notion of lexical similarity that implies also selectional preference for the individual rule arguments. Experiments on a large data set show that selectional restrictions, applied conjunctively to all arguments in a pattern, are able to better select correct vs. incorrect cases. As the designed learning process is completely unsupervised and widely applicable, the method provides a very useful tool for different application and domains. It expresses rules as necessary conditions for equivalence or entailment to hold in unseen texts.

Keywords

Lexical Acquisition, Selectional Patterns, Semantic similarity, LSA

1 Introduction

Textual inference is a key component of many natural language processing tasks. For example, question answering needs inference to find non-trivial answers to general questions. Given the question *"Who played the final of the World Cup?"*, the answer *"Italy"* could be retrieved from the snippet *"Italy has won the final of the World Cup"*, by knowing that the pattern *"X win Y"* entails *"X play Y"*. This type of inferences at the textual level have been successfully exploited in information extraction [13] and question answering [3], and have been recently modeled in the Recognizing Textual Entailment (RTE) challenge [1], where systems are compared on the task of recognizing if a text fragment entails another.

The RTE challenge revealed that system for RTE strongly need knowledge at the linguistic level. In particular, most useful are paraphrase and entailment resources containing lists of entailment rules such as "*X win Y*" $\Rightarrow$ "*X play Y*". While these resource already exist, for example DIRT [6] and TE/ASE [14], they suffer two major limitations which makes their use in inference tasks still a challenge: they lack directionality (i.e. they contain *inference rules* $p \approx q$, where the direction of the entailment between pattern p and q is not known) and they are not accurate enough.

Recent trends in RTE reveal that the second limitation is more critical than the first. Indeed, most cases of textual entailment in real applications are pure paraphrases [2], and then are not much sensitive on directionality. On the contrary, the accuracy issue is fundamental: resources are both too noisy (for example DIRT has an average precision of 0.50 [6]), and too generic. In particular, inference rules are often too generic to be successfully used in application, as they do not indicate explicitly in which context they can be applied. For example, the rule "*X win Y*" $\approx$ "*X play Y*" is useful in the previous example, but also implies the incorrect inference *"Gilmour played guitar in Pink Floyd"* $\approx$ *"Gilmour won guitar in Pink Floyd"*.

Recently, [9] proposed a method to produce more specific rules, in which the admissible arguments for the inference rules are explicitly indicated. For this purpose they use *inferential selectional preferences* (ISP) over the DIRT rules, producing inference rules augmented with selectional preferences (SP). In the above example, the rule would be: $< player >$ play $< competition > \approx < player >$ win $< competition >$. We call these augmented rules *restricted inference rules* (RIR). The two SPs are inferred in [9] as the most common generalization of the X and Y slot fillers in taxonomies such as WordNet or CBC [10]. Yet, this approach suffers from three main problems:

- performance are still low for applications: the ISPs are able to filter correct/incorrect instances of a RIR with 0.59 accuracy;
- it needs pre-existing resources. This makes the method sensitive to the accuracy and the coverage of the resources themselves.
- the computational cost for building the ISPs is high.

In this paper we present a new approach to induce RIRs, based on a LSA-based geometric similarity model. Given an input inference rules $p \approx q$, our algorithm extracts all the X's and Y's slot fillers for p and q in a given corpus. Then, independently for p and q, it derives a vectorial representation in a reduced LSA space for each slot filler. Slot fillers of a given pattern's

slot (e.g X in $< X, p, Y >$) are then clustered according to their similarity in the LSA space. Centroids of these clusters are good candidates to represent the slot's SPs. Satisfiability of SPs is thus mapped into a similarity estimation problem. SPs are then used to build the final rule. Our methods provides the following main contributions:

- the computation of similarity and the clustering in the LSA space offers an effective way to capture text semantics, as LSA is sensitive to both first and second order relations;
- external resources are not needed: SPs are created directly from the textual corpus, thus reducing validation costs and coverage problems;
- complexity is also kept limited, as all similarity computations are done in the reduced LSA space.

In the rest of the paper, we will report empirical evidence to support these claims. In Section 2, we analyze some previous work related to our research; in Section 3 we describe our approach, while in Section 4 we report on the acquired experimental evidence. Finally, in Section 5 we draw final conclusions anf future work.

2 RelatedWork

Automatic methods for acquiring inference rules mainly use pattern distributional properties to infer a similarity score for the relation $p \approx q$. [6] introduce DIRT, a database inference rules, created by first extracting patterns $< X, p, Y >$ from a large corpus using a dependency parser and then creating inference rules $< X, p, Y > \approx < X, q, Y >$as those pairs of patterns p and q which are distributionally similar (i.e. they have similar slot fillers for X and Y). DIRT has an average precision of 0.50 on the task of acquiring inference rules over the top scoring 40 rules for each pattern. [14] present a scalable Web-based approach for inference rule acquisition. Starting from a verb lexicon, the method automatically acquires from the Web useful slot fillers, which are in turn used to discover distributionally similar patterns. The method achieves a precision of 0.44. Other resources for textual inference mainly focus on paraphrasing. The extraction of paraphrase patterns is usually achieved by using aligned/comparable corpora : in [8] Finite State Automata are used both to extract and generate paraphrase using multiple translations of the same story, while in [12] Named Entities are used to locate and drive paraphrase extraction from news about the same story.

Automatic methods for acquiring selectional preferences have been firstly introduced in [11], and have been later exploited in many NLP fields, to restrict the applicability of a given predicate to a pre-defined set of semantic classes. These classes are derived either from manually built resources (such as WordNet) or from automatically harvested ones (such as CBC). As outlined in the Introduction, [9] originally exploit selectional preferences to induce refinements over DIRT inference rules. To our knowledge no attempts have been made so far to induce selectional preference for inference rules without the use of an external taxonomy.

In [7] a Latent Semantic Analysis (LSA) model is applied to explore the relationship between lexical cohesion and entailment. Latent Semantic Analysis [5], captures the essential relationships between documents and word meaning, and tries to tackle the problem of the very large number of dimensions. In [7], LSA is used to model cohesion and coherence, two problems closely related to entailment. A generative model of entailment is formulated, in which the training consists of computing cohesion/coherence over labeled proposition-hypothesis pairs and using logistic regression to fit a supervised classifier to the data. Even if the results supports the basic intuition, the model does not directly deal with the directionality of the relation.

3 Automatic Acquisition of Inference Rules

The goal of our model is to automatically induce from an *inference rule* its set of correct *restricted inference rules* (RIRs). The system goes through the following three steps.

In the first step (Section 3.2), given an inference rule $< X, p, Y > \approx < X, q, Y >$, it considers separately the two patterns $< X, p, Y >$ and $< X, q, Y >$. The goal is to find for the slot X and Y of a pattern p, its set of *selectional preferences* SPs CX^p and CY^p, i.e. the typical semantic classes of X and Y for p. For example $< X, play, Y >$ will have CX^p ={*Player, Actor, Musician*} and CY^p ={*Competition, Piece, Composition*}, and $< X, win, Y >$ will have CX^q ={*Football_Player, Player, Person*} and CY^q ={*Competition, Award*}. A single SP will be hereafter indicated with a pedice: e.g. $CY_1^q \in CY^q$ is used in the example to indicate either *Competition*. Our system performs this first step using clustering in the LSA space: the set of SPs are represented by clusters of slot-fillers in the reduced geometric space.

In the second step (Section 3.3), given the sets of SPs for p and q, the system has to find the pairs $< CX_i^p, CX_j^q >$ and $< CY_i^p, CY_j^q >$ of *compatible SPs* for slot X and Y across the two patters, i.e. the similar semantic classes across p and q for which the inference rule is likely to hold. In the example, the compatible SPs for the slot X are <*Player, Player*> and <*Player, Football_Player*>, and for the slot Y are <*Competition, Competition*>. The system performs this step by estimating compatibility between clusters of p and q, as similarity in the LSA space.

In the third step (Section 3.4), the system has to build the final set of RIRs, by leveraging the pairs of compatible clusters discovered in the previous step. For example it could discover <*Football_Player, play, Competition*>≈<*Player, win, Competition*> and <*Player, play, Competition*>≈<*Player, win, Competition*>. Our system performs this last step by conjunctively applying compatibility constraints over the corresponding slot fillers in a pattern pair.

In the rest of this section, before describing in details the above steps, we introduce in Section 3.1 our main idea of using clustering in the LSA space as a means to discover RIRs.

3.1 Using Latent Semantic Analysis for Rule Induction

LSA [5] is an extension of the vector space model based on the *Singular Value Decomposition* (SVD), a matrix decomposition process that creates an approximation of the original word by document matrix, and captures term semantic dependencies. In LSA, the document space is replaced by a lower dimensional document space M_k, called k-space (or LSA space) in which each dimension is a derived concept. M_k captures the same statistical information in a new k-dimensional space, where each dimension represents one of the derived LSA features (or concepts). These may be thought of as artificial concepts and represent emerging meaning components as a linear combination of many different words (or documents). Terms, on their own, are accordingly represented as combinations of the emerging concepts. The similarity between resulting vectors, as measured by the cosine of the resulting angle, has been shown to closely mimic human judgments of meaning similarity and semantic inference. LSA has two main advantages: first, the computation needed to measure similarity is drastically reduced due to the low k dimensional LSA space; secondly, unlike similarity methods in traditional vector spaces, LSA captures second order relations between words.

3.1.1 Leveraging LSA Similarity for discovering RIRs

As outlined in the introduction, the major limitation of inference rule resources such as DIRT, is that they do not specify for which semantic classes a rule holds. In particular the DIRT model [6] exploits the so-called *Extended Distributional Hypothesis*: "*If two patterns tend to occur in similar contexts, the meanings of the patterns tend to be similar*".

In practice, this means that two patterns are similar if they have *enough* slot-fillers in common, as extracted from a textual corpus. Similarity between contexts (i.e. the slot-fillers and the occurrences of the patterns in the corpus) is thus a key notion for rule induction. Yet, once rules have been induced, these originating contexts are neglected. Then, no further lexical constraint is available to decide when to apply a rule to a novel context (e.g. we don't know if the rule $< X, play, Y > \approx < X, win, Y >$ can be applied to "*David Gilmour plays the guitar*" to derive "*David Gilmour won the guitar*").

The main aim of the model proposed here is to capitalize the idea that originating contexts are very informative about *the lexical conditions under which an inference rule can be triggered*. The goal is then to build a DIRT-like resource in which the slot-filler information is preserved. Yet, storing such information is prohibitive, because of the huge number of lexical slot fillers observable in very large corpora. An alternative representation must be then devised. LSA gives the solution, by a synthetic way to represent slot fillers in the reduced M_k space. In M_k the semantics of the slot fillers is preserved, but the dimensionality of the problem is drastically reduced. In particular, rule induction can exploit the similarity between slot fillers of two patterns p and q in M_k, as a source of semantic information. Also, clustering in the LSA space allows to detect SPs for individual slot fillers, which can be used to decide when to apply a rule in a novel context. As clusters are expressed via an inexpensive vector representations, i.e. their centroids c_X and c_Y, satisfiability of a SP can be modeled via a simple similarity constraint. A newly encountered word w satisfies the SP of a slot filler X *iff* it is *enough similar* to a centroid, i.e. iff $sim(w, c_X) > \tau$, where τ is a positive threshold (the same stands for Y).

The LSA space M_k used for our purpose is that obtained from the original space M, composed by the words (including the slot fillers) and the documents of the corpus from which the resource (e.g. DIRT) has been created. In particular, as requested by LSA, documents are further divided in sub-portions (e.g. few sentences in a paragraph) that constitute coherent discourse segments (e.g. a news, a full story, etc.). In the following sections, we describe as how we implemented this idea.

3.2 Selectional Preferences as clusters in the LSA space

In the first step, the algorithm firstly performs the SVD, obtaining a reduced space M_k in which each slot filler is represented by a vector in the space. Given a pattern p, it is then possible to compute the similarity between its slot fillers using Cosine similarity.

Then, for a given pattern p, the algorithm applies a variant of the *K-means* clustering algorithm to separately cluster the X^p and Y^p slot fillers, using their vectorial representation in the LSA space. The variant is based on the QT (*quality threshold*) cluster algorithm [4], that does not require to specify the number of clusters *a priori*. The basic idea is to impose a threshold representing the maximum allowed distance from the centroid of a cluster; if a words falls beyond this distance, a new cluster is created. The core of the algorithm is described by Algorithm 3.2.

The produced sets of clusters, denoted as CX^p and CY^p, are the LSA representations of the SPs for the X^p and Y^p slots of the pattern p. In other terms, every cluster CX_i^p expresses a group of lexical items (i.e. slot fillers) that act as a sinlge semantic class and provides selectional criterium for deciding the correctness of the pattern use in future contexts. We will then assume that each cluster *is* in fact a SP, and make use of similarity between words and clusters as a selectional preference constrain.

For example suppose that the pattern $< X, play, Y >$ has the following slot fillers for the X^p slot: {*McEnroe, Johnny_Depp, midfielder, Gilmour, comedian, footballer, Baggio*}. Ideally, three clusters should be created: CX_1^p ={*McEnroe, midfielder, footballer, Baggio*}, CX_2^p={*Johnny_Depp, comedian*}, CX_3^p ={*Gilmour*}, which should respectively represent the SPs *Player, Actor, Musician*.

Hereafter, given a generic pattern $< X, p, Y >$, we denote with x_i and y_i the slot fillers of X and Y, i.e. $x_i \in X$ and $y_i \in Y$.

Note that a cluster CX_i^p can be possibly made by a single element (e.g. $CX_3^p = \{Gilmour\}$). Such a cluster is called **trivial**:

Algorithm 1 *QT*-clustering

Require: *QT* {Quality Threshold for clusters}
  repeat
    for all slot fillers w **do**
      Select C_x as the best cluster for w.
      if $sim(w, c_x) > QT$ **then**
        Generate new cluster C_w for w
      else
        Accept w into cluster C_x
      end if
    end for
  until No other shift is necessary or the maximum number of iteration is reached

$$\exists! x \in X^p \text{ such that } x \in CX_i^p \tag{1}$$

i.e. $CX_i^p = \{x\}$. As a final operation, the algorithm compute for each cluster a degree of cohesion. The *cohesion* r_i of a cluster C_i^p is intended as the minimal similarity bewteen the centroid c_i^p, and the cluster members. Given a valid similarity function *sim* (e.g. the cosine similarity) between two instances, the cohesion can be easily defined as $r_i = min_{x_j \in C_i^p} sim(x_j, c_i^p)$. However, accordingly, the cohesion of a trivial cluster C_i would be 1, that is too restrictive to adopt for the usually more vague notion of SP. For example the SP $CX_3^p = \{Gilmour\}$ would accept only contexts which have *Gilmour* as slot fillers; we then need to *relax* the constraint from *Gimour* to *Musician*. To do so, we assume that any cluster C_i must be characterized by having a maximal cohesion that does not exceed a given threshold $\tau \in (0, 1)$. The lexical cohesion r_i of a generic cluster C_i^p can be thus formally defined as follows:

$$r_i = min(min_{x_j \in C_i^p} sim(x_j, c_i^p), \tau) \tag{2}$$

Equation 2 expresses the degree of freedom by which a cluster (i.e. a SP) can be used in future predictions, as described in the next section.

3.3 Discovering compatible clusters

The goal of the second step is to find compatible SPs across two patterns p and q of an inference rule. In the LSA space, the problem is then to identify pairs $< CX_i^p, CX_j^q >$ and $< CY_i^p, CY_j^q >$ of *compatible clusters*, i.e. clusters which are likely to represent the same semantic classes. We then need to define a notion of similarity between two clusters, i.e. a notion of *compatibility*.

DEFINITION (*Compatibility between clusters*). Given two clusters related to the same slot, C_i and C_j with centroids c_i and c_j, C_i is **compatible** with C_j, i.e. $C_i \simeq C_j$, *iff* an element $x_k \in C_j$ exists such that $sim(x_k, c_i) \geq r_i$ or vice versa. The notion of compatibility is leveraged in the next step to induce the RIRs. Also, it can be used to detect if a word satisfies the SP expressed by a cluster of a pattern. An incoming word w *satisfies a SP* expressed by a cluster C_i, if w is likely *a member* of C_i, $w \tilde{\in} C_i$. This means that its similarity with the centroid c_i is *close enough* to the cluster's coherence. A tolerance factor $\sigma(r_i)$, can be here used for the following technical definition.

DEFINITION (*Satisfaction of SPs*). An incoming word w *satisfies a SP* expressed by a cluster C_i, *iff*

$$w \tilde{\in} C_i \textbf{ iff } sim(w, c_i) > r_i - \sigma(r_i) \tag{3}$$

where $\sigma(r_i)$ is a monotonic non decreasing function of the tolerance. The higher is the tolerance the lower it should be the threshold of acceptance. A possible definition for $\sigma(r_i)$ is $max(\alpha r_i^n + \beta, 0)$ with parameters α, β and n to be fixed empirically[1]. The above definition of compatibility and satisfaction for SPs allow us respectively to define a model for inducing RIRs (Section 3.4), and to decide if a text fragment (e.g. "*David Gilmour plays the guitar*") is a valid pattern instance (Section 3.4.1).

3.4 Induction of Restricted Inference Rules

An inference rule $< X, p, Y > \approx < X, q, Y >$states that in most contexts q can be used as a good substitute of p. This indicates a generic *relatedness relation* between two patterns, that may eventually result in an entailment or equivalence relation (more specifically, relatedness between two patterns is a *necessary* condition for an entailment or equivalence relation).

We can here define a more precise and restrictive notion of *semantic relatedness*, which takes into consideration compatible SPs on the slot fillers. This notion is at the base of the RIRs definition.

DEFINITION (*Semantic relatedness between patterns*). Given two patterns p and q, they are semantically related, i.e. $< X, p, Y > \simeq < X, q, Y >$, if their slots are described by compatible clusters. More technically, $< X, p, Y > \simeq < X, q, Y >$holds **iff** for the slots X and Y, two cluster pairs, $< CX_i^p, CX_j^q >$ and $< CY_i^p, CY_j^q >$, can be found such that:

$$CX_i^p \simeq CX_j^q \quad \wedge \quad CY_i^p \simeq CY_j^q \tag{4}$$

Equation 4 makes a consistent use of the geometrical constraints on selectional preferences provided by the LSA transformation for the patterns p and q. This realizes an operational model of ISPs as in [9]. Yet, it has following advantages: it does not imply any generalization of the collocational evidences from text, except the similarity estimated in the LSA space; it does not rely on external resources like WordNet or CBC; it is largely applicable. The above definition of semantic relatedness is used to induce the RIRs.

DEFINITION (*Restricted inference rules*). Given two patterns p and q that are semantically related, every cluster 4-tuple $< CX_i^p, CX_j^q, CY_i^p, CY_j^q >$ satisfying Equation 4 establishes a restricted inference rules:

$$< CX_i^p, p, CY_i^p > \simeq < CX_j^q, q, CY_j^q >$$

This rule justifies the semantic relatedness through the conjunctive satisfaction of all SPs, via the cluster compatibility notion.

[1] A setting, employed after estimation over the development set, is $n = 1$, $\alpha = 0.267$ and $\beta = -0.0167$.

Given two patterns p and q several restricted inference rules can be derived, as Equation 4 can be satisfied by multiple choices of cluster pairs $< CX_i^p, CX_j^q >$ and $< CY_i^p, CY_j^q >$. These different RIRs can be studied to establish some regularities in evoking independent word senses for the support verbs of p and q. In principle, independent verb senses p_1, p_2 could generate different inference rules by selecting (through the compatibility between clusters $< CX_i^p, CX_j^q >$ and $< CY_i^p, CY_j^q >$) different senses of the pattern q. We call the set of RIRs for the patterns p and q a *rule set*:

$$SPR(p,q) = \{< p, q, CX_i^p, CX_j^q, CY_i^p, CY_j^q > \text{ satisfying Eq. 4}\}$$

3.4.1 Leveraging RIRs in Textual Inference

A restricted inference rule should predict if, given a triple like $w_x - p - w_y$ as it is found in an incoming sentence, it is a good candidate for the substitution $w_x - p - w_y \simeq w_x - q - w_y$. This establishes a criteria for deciding when and why an inference rule can be used. This property can be defined as follows.

DEFINITION (*Relational Selectional Preference*). The inference $w_x - p - w_y \simeq w_x - q - w_y$ is accepted **iff** a 6-tuple $< p, q, CX_i^p, CX_j^q, CY_i^p, CY_j^q > \in SPR(p,q)$ exists such that the following condition holds:

$$w_x \tilde{\in} CX_i^p \quad \wedge \quad w_y \tilde{\in} CY_i^p \tag{5}$$

4 Empirical Investigation

4.1 Experimental Set-Up

Aims of the experiments is to evaluate the selective power of the acquired selectional preferences on a consistent set of sentences. At this purpose, we obtained from the authors of [9] the same data and trained our model according to the originating TREC corpus used for those experiments. The corpus is a subset of the TREC-9 collection used to feed the LSA preprocessing phase, the pattern extraction and the clustering. The word pairs used in the test to validate the selectional preferences have been extracted from the 1999 AP newswire collection (part of the TREC-2002 Aquaint collection) consisting of approximately 31 million words. The DIRT resource and the 1999 AP newswire collection represents the Corpus processed with LSA algorithm.

In the first step, the corpus is analyzed by the Minipar parser to recognize nouns, verbs and adjectives. Only these syntactic classes are used to generate the LSA term by document matrix. The total number of documents are 491,384 amounting to 2.9 million token corpus. After the exclusion of types occurring less than 3 times in the collection, a dictionary of 529,964 terms has been obtained. The dimension k of the LSA transformation has been set to 100.

The test data consist of a validation and test set each built selecting 10 word pairs for each of the 50 inference rules $< p, q >$, randomly exacted from DIRT. This amounts to the analysis of a set of 500+500=1,000 instances $< x, p, y >$ hand labeled by 0 if $< x, q, y >$ is not acceptable (negative instances) and 1 otherwise (positive instances). Negative and positive instances are balanced.

Pattern	Clusters (slot Y)	Coherence
bring_by	case, lawsuit, action, judge, discrimination	0.79
file_by	lawyer, attorney, prosecution, judge, counsel, defendant, Justice_Department, FBI, prosecutor,	0.85

Table 1: *Clusters obtained for two of the test patterns.*

4.2 Acquisition of Selectional Preferences

Selectional preferences have been acquired through the selection of the fillers x of a given pattern p and its slot X, appearing in the corpus. Only nouns are clustered in this experiment. Given such lexical group X^p, the clustering algorithm results in a set of cluster CX^p given by the coherent subsets of X^p. With a parameter like $QT = 0.45$ we obtained a total of **13,708** clusters (out of 56,238 analyzed slot fillers). On average this amounts to **60** cluster per slot. This ratio falls down to 26 when $QT = 0.25$ is employed, and only 5,998 clusters are built. The plot of the average number of cluster for different numbers of slot filler as found in the corpus is reported in Fig. 1. The plot suggests that the number of cluster does not grow to much with respect to increasing number of originating slot fillers: when thousands of different slot fillers have been found, we still have no more than 10-20 clusters. The compression factor of our method (i.e. 90% at $QT = 0,25$) allows to represent a pattern by storing few representative information (i.e. cluster centroids). This compression suggests that the corpus evidence about a pattern is meaningful to the description of rules and to the modeling of selectional preferences. In fact, LSA produces high similarity values (among members of a cluster) even when a large number of fillers is considered.

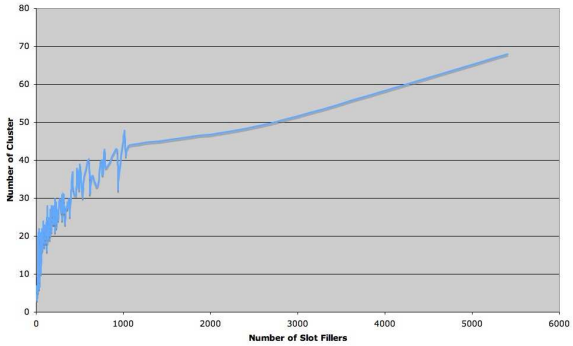

Fig. 1: *Clusters vs. different slot fillers*

An example is shown in Table 1 for the X slot of two different patterns.

4.3 Accuracy of the Inferential Selectional Preference

The performance of inferential selectional preferences are defined via the following scores. Let t^+ represent

Setting	Accuracy	Sensitivity	Specificity
accept all	50.00%	100%	0%
random	49.04%	51%	48.09%
ISP.IIM.or [9]	59%	73%	45%
ISP.JIM [9]	53%	17%	88%
$\tau = 0.7, QT = .45$	54.08%	29.9%	78.6%
$\tau = 0.8, QT = .25$	**61.02%**	61.6%	60%
$\tau = 0.8, QT = .3$	59.3%	55.8%	62.9%
$\tau = 0.9, QT = .25$	**60.83%**	62.5%	59.07%
$\tau = 0.9, QT = .45$	56.5%	60.32%	52.67%

Table 2: *Performance Evaluation of the textual inference rules*

the number of positive instances correctly accepted by the system, t^- represent the number of negative instances correctly refused, f^+ represent the number of accepted negative instances and f^- the number of refused positive instances. *Sensitivity* is defined as $\frac{t^+}{t^+ + f^-}$, i.e. the probability of accepting correct inferences. *Specificity* is defined as $\frac{t^-}{t^- + f^+}$, i.e. the probability of rejecting incorrect inferences. The overall *Accuracy*, i.e. $\frac{t^+ + t^-}{t^+ + t^- + f^+ + f^-}$, captures the quality of pointwise inference over the two classes of instances.

The development data have been firstly used to measure the reachable accuracy over unseen data by also systematically estimating the following parameters:

- maximal distance allowed for cluster acceptance (parameter QT). Different values have been tested ranging from 0.25 to 0.45
- coherence threshold, τ, ranging in $[0.7, 0.9]$
- tolerance function $\sigma(r_i)$, for which two different settings have been tested but $n = 1$ was always the best choice, with small variations for β and α.

The best parameter setting obtained in the development set does not seems to prefer a restrictive strategy in the choice of clusters ($QT = .25$ was the best setting) when more constrained coherence tolerance is applied ($\tau = 0.8 - 0.9$). We then measured the accuracy, significance and specificity over the unseen instances of the test set, according to the best choices (e.g. $\tau = 0.8 - 0.9, QT = .25, n = 1$). Table 2 reports the achieved performance, compared with the baseline (i.e. the "accept all instances" rule), a random choice function and the best joint and independent models, presented in [9]. Figure 2 reports the ROC analysis of the results according to different parameter settings over the test set.

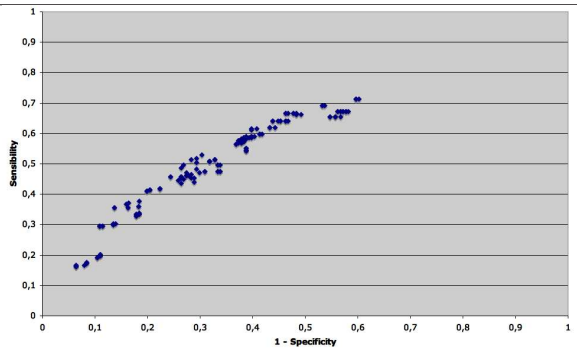

Fig. 2: *ROC analysis over the Test Set*

5 Conclusions

The results reported in Section 4 are more than promising. The simple clustering method proposed in this paper, coupled with the LSA treatment of the source corpus, seems very effective in capturing the local semantics of inference rules. The proposed rule representation extends previous proposals, by exploiting the simple geometrical constraints provided by cluster centroids. The observed compression is a key factor as it confirms the overall consistency of the method. More empirical evidence has to be gathered over other collections, and different clustering techniques and measure must be explored to better assess the proposed acquisition process. This will be part of future research on this topic.

References

[1] R. Bar Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. In B. Magnini and I. Dagan, editors, *Proceedings of the Second PASCAL Recognizing Textual Entailment Challenge*, Venice, Italy, 2006. Springer-Verlag.

[2] S. Bayer, J. Burger, L. Ferro, J. Henderson, and A. Yeh. MITRE's submissions to the eu pascal rte challenge. In *Proceedings of the 1st Pascal Challenge Workshop*, Southampton, UK, 2005.

[3] S. Harabagiu and A. Hickl. Methods for using textual entailment in open-domain question answering. In *Proceedings of ACL 2006*, Sydney, Australia, 2006.

[4] K. S. Heyer, L.J. and S. Yooseph. Exploring expression data: Identification and analysis of coexpressed genes. *Genome Research*, (9):1106–1115.

[5] T. Landauer and S. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240, 1997.

[6] D. Lin and P. Pantel. DIRT-discovery of inference rules from text. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, San Francisco, CA, 2001.

[7] A. Olney and Z. Cai. An orthonormal basis for entailment. In *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference*, pages 554–559, Menlo Park, CA, May 15-17 2005. AAAI Press.

[8] B. Pang, K. Knight, and D. Marcu. Syntax-based alignment of multiple translations: extracting paraphrases and generating new sentences. In *Proceedings of HLT/NAACL-03*, pages 49–56, Edmonton, Canada, 2003.

[9] P. Pantel, R. Bhagat, B. Coppola, T. Chklovski, and E. Hovy. Isp: Learning inferential selectional preferences. In *Proceedings of HLT/NAACL 2007*, 2007.

[10] P. Pantel and D. Lin. Discovering word senses from text. In *Proceedings of KDD-02*, pages 613–619, Edmonton, Canada, 2002.

[11] P. Resnik. *Selection and Information: A Class-Based Approach to Lexical Relationships*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1993.

[12] Y. Shinyama, S. Sekine, K. Sudo, and R. Grishman. Automatic paraphrase acquisition from news articles. In *Proceedings of HLT-02*, San Diego, CA, 2003.

[13] K. Sudo, S. Sekine, and R. Grishman. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of ACL 2003*, 2003.

[14] I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. Scaling web-based acquisition of entailment relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcellona, Spain, 2004.