Feature Distance-based Framework for Classification of Low-Frequency Semantic Relations

André Kenji Horie
School of Information Science and Technology
University of Tokyo
Tokyo, Japan
Email: andre@mi.ci.i.u-tokyo.ac.jp

Mitsuru Ishizuka
School of Information Science and Technology
University of Tokyo
Tokyo, Japan
Email: ishizuka@i.u-tokyo.ac.jp

***Abstract*—In the relation extraction of semantic relations, it is not uncommon to face settings in which the training data provides very few instances of some relation classes. This is mostly due to the high cost of producing such data and to the class imbalance problem, which may result in some classes presenting small frequencies even with a large annotated corpus. This work thus presents a semi-supervised bootstrapped method to expand this initial training dataset, using pattern matching to extract new candidate instances from the Web. The core of this process uses a multiview feature distance-based framework, which allows quantitative and qualitative analysis of intermediate steps of the process. Experimental results show that this framework provides better results in the relation classification task than the baseline, and the bootstrapped architecture improves the relation classification task as a whole for these low-frequency semantic relations settings.**

***Keywords*—Semantic Computing; Concept Description; Natural Language Text**

I. Introduction

The extraction of semantic relations from natural language texts has been of increasing interest in Semantic Computing, since it allows several compelling applications and services to be developed, such as semantic indexing and searching. In order to extract these relations, two approaches are the most evident ones. The first one is to use heuristics to abstract meaning. However, determining these heuristics proves itself to be an arduous task and maybe even infeasible, due to the richness of semantics. The other one is to use machine learning algorithms to analyze semantically-annotated training data, utilizing the generated model to find relations in the testing data, which is a much more reasonable effort.

This annotated training data is acquirable by one of two methods: manually constructing examples specifically for this task, or annotating a whole corpus. While the first suffers from excessive simplicity of the constructed sentences, the latter presents the class imbalance problem, which results in any inherently rare relation class producing very few training instances. Adding the fact that annotating data is a costly process, the classifier will often have to deal with relation classes whose frequency is too low, which ultimately produces incorrect classification due to unknown features in the testing data.

This work thus proposes a bootstrapped architecture, which utilizes a semi-supervised method to expand datasets with low frequency relation classes that are to be used as training data in a semantic relation extraction process. The relation classification of the identified candidate relations for this task is carried by a core framework, which is based on the distance between relation features. This framework combines the distinct views introduced by different feature types into one single multiview matrix, allowing graphical and numerical analysis of intermediate steps of the classification process.

This article is structured as follows. In section II, some theoretical aspects and related work are introduced. In section III, the architecture of the bootstrapped process is outlined. In section IV, the core framework is proposed. In section V, experiments are conducted and results are discussed. Finally, in section VI, a conclusion is provided.

II. Background and Related Work

A. Semantic Relations

Semantic relation is defined as any meaningful association among two or more concepts. Considering an irreflexive relation between two linguistic entities, these entities are then denominated head and tail entities of the relation.

The nature of such associations happen in each of the many layers of semantics. In order to better understand the scope of the works related to semantic relations, it is imperative to first properly categorize the relations concerning their nature. Therefore, semantic relations will be divided into three types, each of which corresponding to different granularity levels.

Semantic relations of type 1 are associations at the word level which are dependent on the lexical information of the entities. Ferdinand de Saussure [1] described paradigmatic relations as ones whose entities can occur in the same position within a context, which is the case of the words *today* and *tomorrow* in the sentence "*The game will be today / tomorrow*". The definition of type 1 relations proposed herein is derived from that of paradigmatic relations, but it also adds the property of context-insensitiveness, and allows irreflexive relations to be included. This way, relations of type 1 are characterized by the lexical association between entities, such as in the example below:

Ex.: *Tokyo→Japan (capital of)*

Semantic relations of type 2 are at the sentence level, and are based on the role that each entity has in the semantics of

978-0-7695-4492-2/11 $26.00 © 2011 IEEE
DOI 10.1109/ICSC.2011.9

the sentence. This definition closely follows [1], which defines syntagmatic relations as ones whose entities co-occur within a given context. It is noticeable that these relations are highly dependent on syntax.

Ex.: *walk→home in "I walked home" (destination)*

Relations of type 2 are the target of tasks such as Semantic Role Labeling (SRL) [2], whose two of the most notable examples are the PropBank [3] and the FrameNet [4].

It is important to observe that relations of type 1 can also be expressed as type 2 when they are presented within a context [5]. For instance, the previous example of type 1 relation *Tokyo→Japan* can be found within a sentence, as in "*Tokyo is the capital of Japan*". The lexical connection between head and tail entities is explicitly expressed here by "*is capital of*", and the relation can thus be thought as of type 2, since context, instead of only the interlocutor's previous knowledge, provides lexical information.

Finally, semantic relations of type 3 are at a higher level of text [6] and occur beyond the clause and sentence boundaries. They can be analyzed from a logical perspective, such as in equivalence, contradiction and cause-and-effect relations, or from a textual perspective, such as in cohesion and coherence.

Ex.: *"If it rains, I will not go" (conditional)*

Type 3 relations are studied by tasks such as discourse parsing, which are examplified by the RST Discourse Treebank [7] and the Penn Discourse Treebank [8].

B. Concept Description Language (CDL)

CDL [9] is a language proposed by the institute of Semantic Computing of Japan (ISeC)[1] that describes the concepts expressed in different types of media. The subset of CDL that offers general support for natural languages is officially called CDL.nl [10], but it will be denominated herein as "CDL" for simplicity purposes.

The representation of semantics in CDL is based on entities, relations and attributes. The entities and relations form a directed graph network for a given text, whereas attributes describe some properties of concepts, but these will not be further investigated in this work.

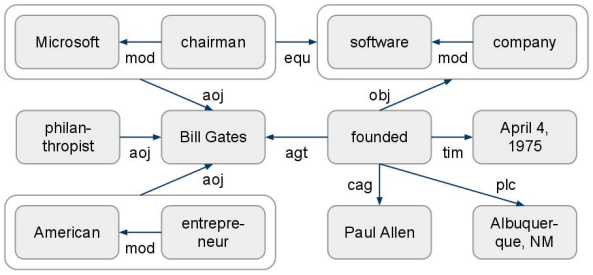
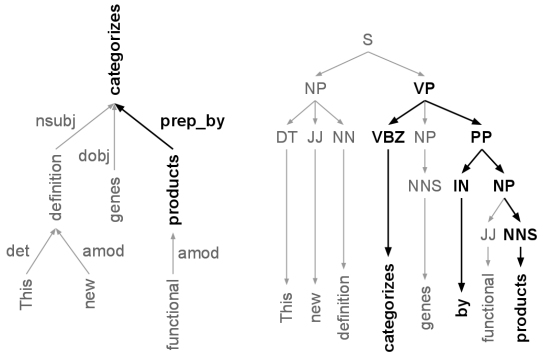


Fig. 1. CDL graphical representation of a sentence

Figure 1 illustrates the sentence "*Bill Gates is an American entrepreneur, philanthropist and chairman of Microsoft, the software company he founded with Paul Allen in Albuquerque, New Mexico, on April 4, 1975*" annotated using CDL and shown through its graphical representation. The semantic relation classes proposed by CDL are mostly of type 2, as is the case of the Agent (agt) and Object (obj) relations. However, some of them are of type 1, as in the Equivalence (equ) relation, and even of type 3, such as the Conditional (con) relation, which is not stated in the example.

Nevertheless, since relation classes of type 1 in the case of CDL are always expressed within a context, they can also be seen as a type 2, for the reasons stated in II-A. In addition, since relation classes of type 3 for CDL explicitly state the discourse connector, they are highly dependent on the syntactic structure, just like relations of type 2. Because these classes do not have the complexity observed in other discourse parsing tasks, the same detection and classification methods can be used for all CDL relation classes, maintaining a single method for simplicity purposes, as evaluation of discourse parsing techniques is not the intent of this work.

Just as many of the semantic relation classification schemes, CDL also presents inherently rare classes. The observed class frequencies in a data source generated from Wikipedia are stated in table I. This data source consists of relation instances annotated for nine Wikipedia articles, presenting very few instances for many of the relation classes.



(a) Dependency tree (b) Phrase structure

Fig. 2. Shortest paths for the syntactic features

C. Modeling Semantic Relations

For various tasks concerning semantic relations, the relations are modeled by morphological, syntactical and lexical-semantic features. Some of the recurring features types are:

- Head and tail part-of-speech (POS) tag: Morphological information that describes the class of head and tail entities. Ex.: categorizes/VBZ (present tense verb, third person singular)
- Dependency tree shortest path: The dependency tree provides the grammatical relations among words. By using the shortest path [11] between head and tail entities, it is possible to extract the part of the tree that is relevant to the relation. Figure 2a illustrates the shortest path of a tree using Stanford dependencies.

[1]http://www.instsec.org/ (in Japanese)

TABLE I
FREQUENCIES FOR CDL RELATION CLASSES IN THE WIKIPEDIA DATASET

Class	Frequency	Class	Frequency	Class	Frequency	Class	Frequency	Class	Frequency	Class	Frequency
agt	874	cob	0	icl	0	obj	3339	pos	165	src	65
and	1185	con	8	ins	3	opl	4	ptn	14	tim	124
aoj	2399	coo	0	int	0	or	213	pur	165	tmf	8
bas	17	dur	37	iof	38	per	2	qua	326	tmt	4
ben	17	equ	31	man	863	plc	186	rsn	40	to	37
cag	2	fmt	9	met	28	plf	0	scn	54	via	4
cao	0	frm	20	mod	1807	plt	1	seq	1		
cnt	45	gol	120	nam	9	pof	14	shd	0		

- Phrase structure shortest path: Phrase structure provides syntactic information of a sentence by breaking it into constituent parts (phrasal categories). Figure 2b illustrates the shortest path for the phrase structure of a sentence using the Penn Treebank notation.
- Head and tail named entity (NE) tag: Lexical information that indicates proper nouns, labeling them as people, institutions or places, of the head and tail entities. Ex.: New York/PLACE
- Head and tail WordNet sense: WordNet [12] is a lexical database for English that provides word senses for head and tail entities. The senses are structured in a tree-like structure. Ex.: <verb.cognition> categorize#1

D. Extraction Tasks for Semantic Relations

The semantic relation extraction task for the CDL relations was introduced in [13]. It proposed a hybrid method with a rule-based relation detection and a feature-based relation classification step. For the rule-based detection, heuristics based on the syntax of a sentence are defined, and candidate relations are detected from the corpus. As for the feature-based relation classification, a feature vector (i.e. a vector v in which each element v_k indicates the existence of each feature k) is extracted for each training and candidate relation instances. The feature vectors of the training instances are used to build the SVM classifier model, which in turn is used to classify the vectors for the candidate instances. The experiments showed that while the classification step produced satisfactory results for high-frequency classes, the results for the detection step were below expectation.

For dealing with the small training data setting, we can mention two works. In the first one, a bootstrapped set expansion architecture with a graph-based method was proposed by [14] for type 1 relations. Given the dual behavior of such relations [15], the process starts with two initial sets, one of entity pairs such as *Tokyo→Japan* and the other of contexts such as "*is capital of*", which are then used to expand each other using bootstrapped Web search and filtering. Finally, the results are ranked according to their relevance to the initial seed using ranking score propagation on the intra-view entity pair and context graphs, and the inter-view correlation graph.

The second one, proposed by [16], is a feature vector extension method for type 3 relations. Given a d-dimensional feature vector $v^i = [v^i_1, \ldots, v^i_d]$, $v^i_j \in \mathbb{R}$, for each relation instance i, it calculates a feature co-occurrence matrix C generated using the χ^2-measure on features extracted from a large unlabeled dataset, and uses this matrix to add new elements to the initial feature vectors v^i. This extended features represent correlation among features inexistent in the training data.

The differences in the nature of type 2 relations compared to types 1 and 3 require another approach to relation modeling, and thus to the relation classification task. This is accomplished herein by combining several methods, including Web extraction of sentences possibly containing candidate relations, candidate relation identification using syntactic pattern matching, and classification using feature distance, in order to improve the overall performance of the relation classification task proposed by [13] in the cases in which the amount of training data would otherwise compromise the results.

III. ARCHITECTURE

A. Overview

The architecture for improving the training dataset proposed herein is based on bootstrapped set expansion, similarly to [14]. However, due to the nature of the type 2 semantic relations, it is expected that the Web search generates a lot of noise. In addition, syntax-based pattern matching must be carried, instead of using sequential pattern mining algorithms [17] such as prefixspan [18], and most importantly, a novel multiview feature-based method becomes necessary.

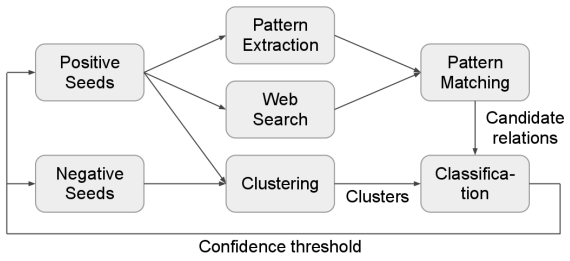


Fig. 3. Architecture of the bootstrapped set expansion process

The overview of the architecture is given in figure 3. First, given a certain relation class R_N, consider positive and negative sets of relation instances. Then, sentences that may contain a relation instance similar to one of the positive relations are extracted from the Web. In the same time, patterns are also extracted from these positive relations. These patterns are then matched against the extracted sentences, generating candidate relations. These candidates are classified using positive and

negative examples, and are fed as seeds for the next iteration of a bootstrapped process. In the end, it is expected that the set of positive relations is larger than the original one, but with recombined and new features.

B. Detailed Architecture

The process starts with a seed data which consists of a relation set R, composed of many relation classes R_N, so that $\bigcup_N R_N = R$ and $R_{N1} \cap R_{N2} = \emptyset, \forall R_{N1}, R_{N2}$. For each relation class R_N, we define the set of positive relations R^+ and the set of negative relations R^- to be used in a one-vs-all relation classification. We can consider R^+ to be all instances whose class is R_N (i.e. the class to be evaluated), and the R^- to be just a subset of the remaining relations. In fact, only relations that are syntactically or semantically similar to R^+ are used for R^-, based on the assumption that non-similar relations are considered easily separable by a classifier. Class similarity is information available in the specification in the case of CDL [10].

From this seed data, candidate relations are extracted through the following steps: Web search, pattern extraction and pattern matching, as stated previously. For the Web search step, the positive relations R^+ are used to generate two Web search engine queries, one of which substituting the head entity for a wildcard, and the other substituting the tail entity. Moreover, all prepositions and conjunctions that occur between the head and tail entities in the phrase structure are considered. For instance, the instrument (INS) relation *categorizes→products* in the sentence "*The new definition categorizes genes by functional products*" produces the following queries:

(1) "categorizes by *"
(2) "* by products "

The pattern extraction step consists of identifying patterns from the positive relations R^+. For this matter, syntactic patterns are used, since the semantic relations in our scope are highly dependent on the syntactic structure. The patterns considered are based on the dependency tree shortest path (figure 2a), since they generate reasonably less noise than those based on phrase structure, increasing the precision of the results. For the same example, the pattern would be as follows:

<*head*> [*prep_by*] <*tail*>

Finally, for the pattern matching step, the Web search results of the first step are matched against the patterns extracted in the second step. For example, if the following two sentences resulted from the Web search engine query, then relation (A) would be matched by *categorizes→products*, but relation (B) would not, since the head entity of (B) is a noun, not a verb.

(A) *interact→Armed_with_Science* in the sentence "*... interact with Armed with Science...*"
(B) *relationships→patients* in the sentence "*Improving relationships with patients*"

After candidate relations are extracted, they are classified using a confidence measure. Given two sets of positive and negative relations R^+ and R^-, let $\mathcal{C}$ be a classifier that creates a classifying model. This model is a function $f : R \rightarrow [-1, 1] \in \mathbb{R}$ that for each input relation instance r_i, produces the confidence measure θ' of this classification. This measure θ' assumes a value of -1 if the classifier is absolutely certain that $r_i \notin R^+$, $+1$ if it is absolutely certain that $r_i \in R^+$, or any intermediate values between these. How this classification is carried is presented further in section IV.

Only the candidates whose classification confidences θ' are larger than a specified threshold θ are used as seeds of the next iteration of this bootstrapped process. Using a semi-automatic process to avoid error propagation in the bootstrapping iterations is advisable if no errors are tolerable.

IV. Core Framework

A. Overview

The core framework performs classification of relations in situations in which the training dataset R presents very few instances. Given the positive and negative relation sets R^+ and R^- of a relation class R_N, it provides one-versus-all classification with a confidence measure output.

The feature distance-based method was chosen as it is very suitable to address the unknown feature problem. However, in order to properly account for the importance of each different feature type, a multiview approach is proposed. This not only provides a way to cluster the relation instances in the training data, but it also allows qualitative and quantitative analysis of the unified view and of each feature type separately. This is a much desired property especially for the rare relation class setting, in which it is crucial to understand how the training data responds to the feature set and how the feature set describes the training data.

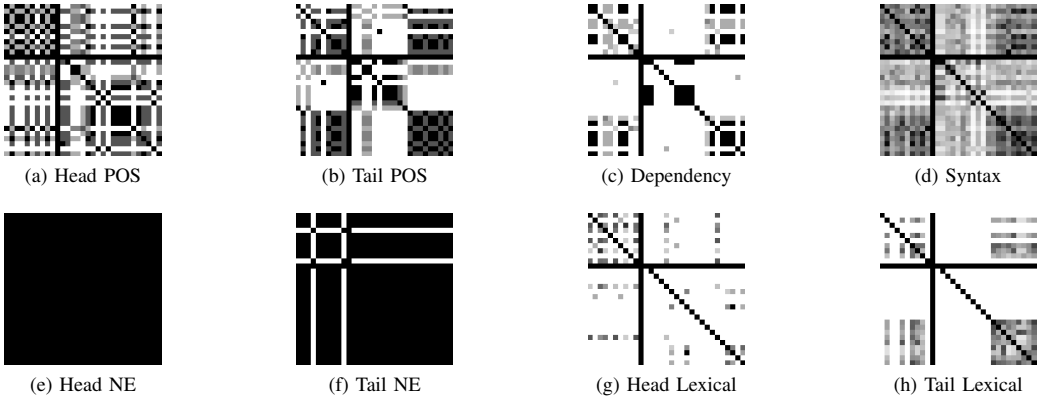
B. Single View Distance Matrix

When classifying low-frequency semantic relations, one of the largest concerns is when features of the testing data are unknown to the classifier. One way to address this problem is to guarantee that there will always be a value to be used in the classification.

For each feature type k, we define the distance measure $\delta_k : R \times R \rightarrow \mathbb{R}$ as a function that outputs a value in the range $[0, 1]$ for any two input relation instances. Obviously, $\delta_k(r_i, r_i) = 0, \forall r_i \in R$.

A list of distance measures for the eight feature types used in this work is presented below:

- Head and tail POS tags: Pre-defined distances among POS tags
- Dependency tree shortest path: Normalized general Levenshtein distance
- Phrase structure shortest path: Normalized general Levenshtein distance
- Head and tail NE tags: Binary (0 if equal, 1 otherwise)
- Head and tail word senses: Given s_1, s_2 word senses, cp their common parent, r the root of the lexical hierarchy tree, and $d(n_1, n_2)$ the number of edges between tree nodes n_1 and n_2, then $\delta = \frac{min(d(s_1,cp),d(s_2,cp))}{d(cp,r)+min(d(s_1,cp),d(s_2,cp))}$



(a) Head POS (b) Tail POS (c) Dependency (d) Syntax

(e) Head NE (f) Tail NE (g) Head Lexical (h) Tail Lexical

Fig. 4. Graphical representation of distance matrix for the INS relation class

Having defined the distance measures, we then define the single view distance matrix D_k as an $n \times n$ matrix, $n = |R|$, as the matrix of the distances between every two relations instances of R:

$$D_k = [\delta_k(r_i, r_j)]_{n \times n}, \quad r_i, r_j \in R \tag{1}$$

Figures 4a through 4h illustrate the single view distance matrices for the INS relation class. This graphical representation is on greyscale, where darker pixels represent lower distances (closer to 0). A black line was also drawn to better separate positive (R^+) and negative (R^-) relations.

C. Multiview Distance Matrix

Each distance matrix D_k provides only one view of the classification problem. However, a consolidated view of all distance matrices is desirable, but this is not trivial to obtain, since it is difficult to account for the importance of features that measure different properties. One possible approach is to consider a linear model:

$$\mathcal{D} = \beta_0 + \sum_k \beta_k \cdot D_k \tag{2}$$

The $n \times n$ matrix $\mathcal{D}$ is called the multiview distance matrix, and is an integrated view of all D_k matrices. It uses linear coefficients β, which indicate the weight of each feature type.

In order to find $\mathcal{D}$, we first need to find β coefficients for the hypothetical optimal case. Let the $n \times n$ matrix Y be the multiview distance matrix for this case. In this situation, considering that the relation instances in R can be grouped in hard clusters $C_i \subset R$, we have that:

$$Y_{ij} = \begin{cases} 0 & \text{if } r_i \in C_i,\ r_j \in C_j \text{ and } C_i = C_j \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

This leads to a block diagonal matrix if the relation instances are sorted by cluster. However, it is not possible to know beforehand the number of clusters in Y. A best-effort approach is then taken at this step, considering that each relation class generates only one cluster. Proper clustering will be carried in future steps. Figure 5a illustrates an example of matrix Y.

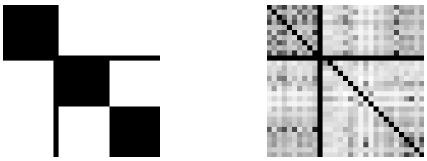
Since Y is a symmetric matrix and values of Y_{ij} for i, j between $|R^+| + 1$ and $|R^+| + |R^-|$ do not matter for the classification in the set expansion setting, the number of equations m can be decreased. As a result, considering equation 2, the following system of equations is observed:

$$Y_{ij} = \beta_0 + \sum_k \beta_k \cdot D^k_{ij}, i = 1 \ldots |R^+|, j = i + 1 \ldots n \tag{4}$$

This can be rewritten using an $m \times k + 1$ matrix D' and an m-dimensional array Y', which contain the values of D and Y from equation 4 respectively, as follows:

$$\beta = (D'^T D')^{-1} D'^T Y' = D'^+ Y' \tag{5}$$

The β coefficients can now be easily calculated using least squares multiple regression. Although the algorithm involves a step of singular value decomposition (SVD), which has a time complexity of $O(m(k+1)^2)$, this should not be an issue, since only eight features are used and the number of relation instances is greatly restricted.



(a) Y (b) $\mathcal{D}$

Fig. 5. Multiview distance matrices Y and $\mathcal{D}$ for the INS relation class

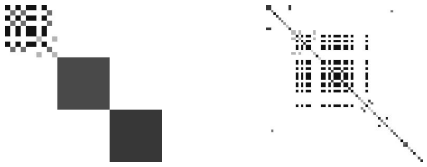
The multiview distance matrix $\mathcal{D}$ is now calculated using equation 2 and the result of equation 5. Figure 5b illustrates $\mathcal{D}$ when calculated from the single view matrices in figure 4.

D. Clustering and Classification

In this work, three different methods for clustering of $\mathcal{D}$ and classification of candidate relations will be used for comparison purposes: a spectral clustering with distance-based classification, a hybrid spectral clustering and SVM

classification, and the baseline SVM classification. For each classification, a confidence measure will also be proposed, as this is required for the bootstrapped process.

Spectral clustering [19] uses the spectrum of the distance matrix in order to perform dimensionality reduction. The algorithm has as input α^*, which controls the recursive partitioning of the algorithm. The value for α^* is found by grid search, testing which of several values produces the best clustering, that is, the clustering that produces a result as close to a block diagonal as possible. Figure 6 presents examples of good and bad spectral clustering results of different distance matrices.



(a) Good clustering (b) Bad clustering

Fig. 6. Examples of spectral clustering

The relation classification for this clustering will be distance-based. The distance between a relation instance r_i and a cluster C_j is given by:

$$\delta(r_i, C_j) = min(\delta(r_i, r_j)), \quad \forall r_j \in C_j \tag{6}$$

The class assigned by the classifier corresponds to the cluster with lowest distance C. In addition, the confidence measure is given by:

$$\theta' = 1 - \frac{\delta(r_i, C)}{\sum_j \delta(r_i, C_j)} \tag{7}$$

The second clustering and classification algorithm is a hybrid spectral and SVM [20]. The clusters are found in the same way as the normal spectral clustering. However, the difference lies in the input of the SVM. Case (1) below illustrates the input using the feature vector v^i for a relation instance i, and (2) using the result of the spectral clustering:

(1) $<+1|-1> : <v_1^i> <v_2^i> <v_3^i> \ldots$
(2) $<C_i> : <\mathcal{D}[i,1]> <\mathcal{D}[i,2]> <\mathcal{D}[i,3]> \ldots$

The confidence measure for SVM classification is a sigmoidal probabilistic output [21].

Finally, the third classification algorithm is the baseline SVM proposed by [13], which uses feature vectors as input. The original method will be extended by adding the sigmoidal probabilistic output confidence measure.

V. Experiments and Results

A. Core Framework Evaluation

In order to analyze the behavior of the core framework, the classification of newly acquired relations for one iteration of the bootstrapped process is evaluated using macro-average accuracy and precision. Accuracy measures the percentage of the correct classifications, whereas precision measures the percentage of correct positive classifications. Macro-average stands for an average which gives equal weights for each relation class R_N, instead of giving equal weights for each individual relation instance. The two metrics are expressed in equations 8 and 9 below:

$$\text{MAAcc} = \frac{1}{|R_N|} \sum_{R_N} \frac{\text{Correct predictions for } R_N}{\text{Total predictions for } R_N} \tag{8}$$

$$\text{MAPrec} = \frac{1}{|R_N|} \sum_{R_N} \frac{\text{Correct positive predictions for } R_N}{\text{Total positive predictions for } R_N} \tag{9}$$

It is noticeable that the recall measure is not used in this context, since the percentage of positive instances identified from the total instances is not important. The extracted positive cases classified as negative may be ignored without further losses to the process.

The newly acquired relations mentioned previously are candidate relation instances that have been confidently classified. Given a threshold value θ, a confidently classified relation has a confidence measure θ' for a given classification method such that $\theta' \geq \theta$. The accuracy and precision values are then calculated for different values of $\theta \in [0, 1]$. They are also calculated in function of the percentage of newly acquired relation instances that are confidently classified.

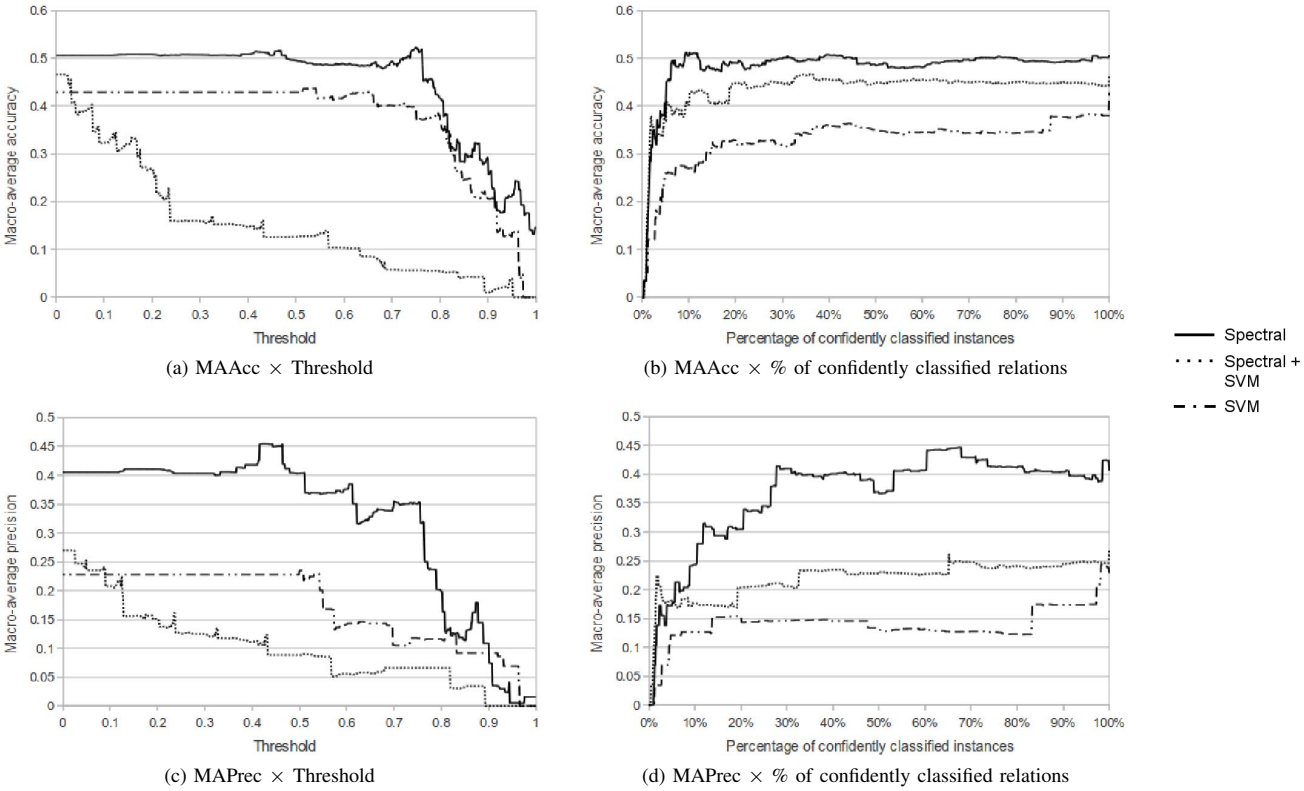
The training data for this task is a small subset of the Wikipedia-annotated corpus presented in section II. For each one of the 29 CDL relation classes that presents at least one similar class and at least one relation instance in the training data, a set of at most 10 instances is randomly selected.

The results for this experiment are given in figure 7. Spectral clustering with distance-based classification provides the best results for all situations, and greatly outperforms the other methods when considering precision. This indicates that the SVM classifier from the baseline method is not able to distinguish the positive class in the one-vs-all problem, which strongly suggests the existence of the unknown feature problem. It is also evident in the hybrid method that the confidence measure provides poor ranking, as observed from the disparities in threshold and percentage of confidently classified results graphs.

B. Architecture Evaluation

For the evaluation of the whole architecture, a training dataset was manually constructed. It aims to describe the relations for each class thoroughly using as few instances as possible. As a result, it is composed of an average of 4.544 relation instances per class, and at most seven instances for one given class. The testing dataset is the one presented in section II.

In this experiment, the improvement in performance of the method in [13] after some iterations of the bootstrapped process is analyzed. The performance metrics used are macro-average precision, recall and F-value. After each iteration, a manual classification step is also carried, in order to minimize the effects of error propagation. Moreover, the features used are the ones presented in this work in section II, instead of the ones in the baseline work, since the main objective



(a) MAAcc × Threshold

(b) MAAcc × % of confidently classified relations

(c) MAPrec × Threshold

(d) MAPrec × % of confidently classified relations

Fig. 7. Experimental results

is to compare the difference in the performance of relation classification regardless of which features are considered.

The overall performance per iteration process is given in table II. By analyzing the results, it is observed that a larger training dataset provides better macro-average precision, recall and F-value as expected, since the unknown feature problem is properly addressed and the existing features are recombined.

TABLE II
OVERALL PERFORMANCE PER ITERATION OF THE BOOTSTRAPPED PROCESS

Iteration	MA Precision	MA Recall	MA F-Value	Set size
Initial	29.15%	27.20%	28.14%	209
#01	36.74%	34.97%	35.83%	969
#02	42.55%	35.58%	38.75%	2357

Detailed results for some of the relation classes are stated in table III, in which entries are ordered by the improvement in the F-value for the first iteration. From these results, it is observed that while three relation classes (OPL, CAG and INS) started being identified by the classifier, one of them (PTN) stopped being identified. In addition, increasing the amount of instances was not necessarily benefitial for some individual classes, although the observed net effect is indeed positive, as the majority of the relation classes experience improvement.

It is important to notice that although this work is focused on evaluating the method for CDL relations, it can be applied to the relation classification task of other low-frequency semantic relations that can also be modeled as a vector composed by features that express different dimensions, as is the case of semantic relations of type 2.

VI. CONCLUSION

This work proposed a bootstrapped architecture and a multiview feature distance-based framework in order to deal with syntax-dependent semantic relations, in a setting in which the training data of the relation extraction process presents few instances for some of the relation classes. The proposed framework is used as the core of a bootstrapped architecture, which intends to expand the initial set of training data using a semi-supervised method.

By using feature distances, the framework is able to minimize the effects of the unknown feature problem, which happens when the classifier does not recognize a feature present in the testing data that was not in the training data. Moreover, it provides a consolidated view of the different feature types by proposing a multiview distance matrix. This matrix can be used for qualitative and quantitative analysis of intermediate steps of the classification process, and is especially desirable when the size of training dataset is small, since it becomes possible to analyze how the training data responds to the chosen feature set or how the feature set describes the training data, and even try to predict the future

TABLE III
DETAILED PERFORMANCE AFTER ONE ITERATION

		INITIAL DATASET				ITERATION #01				
Class	Freq	P_0	R_0	F_0	Set size	P_1	R_1	F_1	Set size	$\frac{F_1-F_0}{F_1+F_0}$
OPL	4	0.00%	0.00%	0.00%	3	33.33%	25.00%	28.57%	5	1.00
CAG	2	0.00%	0.00%	0.00%	4	14.29%	50.00%	22.22%	17	1.00
INS	3	0.00%	0.00%	0.00%	4	7.14%	66.67%	12.90%	22	1.00
BAS	17	5.56%	23.53%	8.99%	5	62.50%	29.41%	40.00%	20	0.63
DUR	37	20.00%	8.33%	11.76%	5	37.04%	27.78%	31.75%	24	0.45
RSN	40	4.03%	12.50%	6.10%	5	36.36%	10.00%	15.69%	15	0.44
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
IOF	38	17.65%	7.89%	10.91%	4	100.00%	2.63%	5.13%	6	-0.36
TO	37	40.00%	10.81%	17.02%	4	7.14%	5.41%	6.15%	25	-0.46
PER	2	5.56%	100.00%	10.53%	5	1.80%	100.00%	3.54%	34	-0.49
PTN	20	25.00%	5.00%	8.33%	5	0.00%	0.00%	0.00%	24	-1.00

outcome of the classification process.

Finally, for the experimental results, when using spectral clustering, the proposed method outperforms other clustering and classification approaches, indicating that the core framework provides better support for the low-frequency semantic relation setting. The expanded training dataset obtained by running some iterations of the bootstrapped process is also able to improve the relation classification task as a whole, proving to be a less costly alternative to corpus-annotation.

REFERENCES

[1] F. de Saussure, *Course in General Linguistics*, 3rd ed. Fontana/Collins, 1916, edited by C. Bally and A. Sechehaye, translated by W. Baskin (1970).

[2] X. Carreras and L. Màrquez, "Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling," in *CoNLL-2005: Proceedings of the 9th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2005.

[3] M. Palmer, D. Gildea, and P. Kingsbury, "The Proposition Bank: An Annotated Corpus of Semantic Roles," *Computational Linguistics*, vol. 31, no. 1, pp. 71–106, 2005.

[4] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet Project," in *Proceedings of the 17th International Conference on Computational Linguistics*, 1998, pp. 86–90.

[5] M. W. Evens, B. Litowitz, J. Markowitz, R. Smith, and O. Werner, "Lexical-Semantic Relations: A Comparative Survey," *Linguistic Research*, 1980.

[6] C. S. G. Khoo and J. C. Na, "Semantic Relations in Information Science," *Annual Review Information Science & Technology*, vol. 40, no. 1, pp. 157–228, 2007.

[7] L. Carlson, D. Marcu, and M. Okurowski, "Building a Discourse-tagged Corpus in the Framework of Rhetorical Structure Theory," in *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*, vol. 16. Association for Computational Linguistics, 2001, pp. 1–10.

[8] E. Miltsakaki, R. Prasad, A. Joshi, and B. Webber, "The Penn Discourse Treebank," in *Proceedings of the 4th International Conference on Language Resources and Evaluation*. Citeseer, 2004.

[9] T. Yokoi, H. Yasuhara, H. Uchida, M. Zhu, and K. Hashida, "CDL (Concept Description Language): A Common Language for Semantic Computing," in *WWW 2005: Online Proceedings of the Workshop on the Semantic Computing Initiative (Sec2005)*, Makuhari, Japan, May 2005.

[10] H. Uchida, M. Zhu, and T. D. Senta, *Universal Networking Language*. UNDL Foundation, 2005.

[11] R. C. Bunescu and R. J. Mooney, "A Shortest Path Dependency Kernel for Relation Extraction," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. ACL, 2005, pp. 724–731.

[12] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*. Cambridge, MA; London: MIT Press, May 1998.

[13] Y. Yan, Y. Matsuo, M. Ishizuka, and T. Yokoi, "Annotating an Extension Layer of Semantic Structure for Natural Language Text," in *ICSC 2008: Proceedings of the 2008 IEEE International Conference on Semantic Computing*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 174–181.

[14] H. Li, D. Bollegala, Y. Matsuo, and M. Ishizuka, "Using Graph Based Method to Improve Bootstrapping Relation Extraction," in *CICLing 2011: Proceedings of the 2011 Conferences on Computational Linguistics and Natural Language Processing*, Tokyo, Japan, 2011.

[15] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Relational Duality: Unsupervised Extraction of Semantic Relations between Entities on the Web," in *WWW 2010: Proceedings of the 19th International Conference on World Wide Web*. New York, NY, USA: ACM, 2010, pp. 151–160.

[16] H. Hernault, D. Bollegala, and M. Ishizuka, "A Semi-Supervised Approach to Improve Classification of Infrequent Discourse Relations using Feature Vector Extension," in *EMNLP 2010: Proceedings of the 2010 Conference on Empirical Methods on Natural Language Processing*, Massachusetts, USA, 2010.

[17] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring the Similarity Between Implicit Semantic Relations from the Web," in *WWW 2009: Proceedings of the 18th International Conference on World Wide Web*. New York, NY, USA: ACM, 2009, pp. 651–660.

[18] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-projected Pattern Growth," in *Proceedings of the 7th International Conference on Data Engineering*. IEEE Computer Society, 2001, pp. 215–226.

[19] R. Kannan, S. Vempala, and A. Vetta, "On Clusterings: Good, bad and spectral," Yale University, Tech. Rep., 2000.

[20] T. Joachims, "Text Categorization with Suport Vector Machines: Learning with Many Relevant Features," in *ECML 1998: Proceedings of the 10th European Conference on Machine Learning*. London, UK: Springer-Verlag, 1998, pp. 137–142.

[21] J. C. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," in *Advances in Large Margin Classifiers*, 1999, pp. 61–74.