Classifier-based Pattern Selection Approach for Relation Instance Extraction

Abstract. A classifier-based pattern selection approach for relation instance extraction is proposed in this paper. The classifier-based pattern selection approach proposes to employ a binary classifier that filters patterns that extracts incorrect entities for a given relation, from pattern set obtained using global estimates such as high frequency. The proposed approach is evaluated using two large independent datasets. The results presented in this paper shows that the classifier-based approach provides a significant improvement in the task of relation extraction against standard methods of relation extraction, employing pattern sets based on high frequency. The higher performance is achieved through filtering out patterns that extract incorrect entities, which in turn improves the precision of applied patterns, resulting in significant improvement in the task of relation extraction.

1 Introduction

Pattern-based information extraction systems have focused on extracting entities for specific relations. For example, given a sentence "Mozart was born in 1756" the task for extracting entities for the relation PERSON-BIRTHYEAR is to extract predicates of the form PERSON-BIRTHYEAR*(Mozart, 1756)*. Similarly the triple COMPANY-CEO*(Google Inc., Sundar Pichai)* is extracted from the sentence "Sundar Pichai is the current CEO of Google Inc." for the relation COMPANY-CEO. Several studies have proposed various types of patterns for extracting entities related to such relations. For example, [1] generated patterns using lexical terms between entities. Similarly, [2] and [3] derived patterns by employing lexico-syntactic features such as Part-Of-Speech (POS) tags. Studies have also proposed dependency parse based syntactic features [2–4] and frame-based semantic features [5–8] for IE.

Equally important to the process of pattern learning and entity extraction is the creation of an optimum set of patterns to ensure extraction of correct entities for specific relations. The goodness measures commonly employed to create such optimum set of patterns considers measures such as frequency [9] or accuracy of patterns [10–12]. Filtering patterns employing goodness measures results in a fixed set of ranked patterns [13], which are then used to extract entities for specific relations. However such methods do not adjudge the quality of patterns with respect to the instances extracted by patterns. For example, while a pattern for a given relation, irrespective of its type i.e., whether it is lexical, syntactic or semantic extracts correct instances from sentence s, the same pattern may extract wrong instances from a different sentence s'. For instance consider the following example sentences :

1. The CEO of the company, **Steve Jobs** *announced the products of* **Apple** at WWDC.
2. Today, **Amazon** *announced the products of* **Apple** on their website.

In the example sentences above, while the lexical pattern "announced the products of" when applied on Sentence 1, extracts correct entities *(Steve Jobs, Apple)* for the relation CEO-COMPANY, the same pattern, when applied on Sentence 2, extracts incorrect entities for the relation CEO-COMPANY *(Amazon, Apple)*. Thus, it is difficult to adjudge patterns by simply considering the number of times the pattern extracts correct and incorrect instances. Further, the fixed set of patterns used for entity extraction is often created independent of the sentences on which the patterns are applied. This implies that none of the useful local information from the target sentence is considered before applying the pattern. For instance, in the example Sentence 1, the terms *CEO* and *WWDC* can serve as useful indicators to extract arguments for the relation CEO-COMPANY. However, such information is not considered before applying the pattern.

Contrary to the approach of fixed set of patterns, this paper presents a classification-based pattern selection approach for relation instance extraction. The classification-based approach proposes to employ a binary classifier that filters patterns that extract incorrect entities from the sentence for a given relation, from the large pattern set obtained using global estimates such as high frequency patterns. Thus, the classifier is useful for (a) selecting a subset of patterns that often extracts correct instances from the sentence; and (b) improving accuracy for pattern based relation instance extraction by reducing incorrect extractions.

More specifically, the key contribution of this paper is a binary classifier that is trained to determine whether a pattern should be applied on test sentences. A seed set of relational instances is used to automatically generate positive and negative training instances for the classifier, thereby minimizing the manual effort required to build the classifier. The classifier is evaluated against employing fixed pattern sets created using global estimates such as frequency. Further, the experiments are conducted on two independent datasets: (a) Wikipedia dataset, developed following distant supervision assumption [14]; and (b) Riedel et al. (2010) dataset [15], which is developed by relaxing the distant supervision assumption.

The remainder of this paper is organised as follows. In §2, we describe the related work to this study. In §3 the proposed classification-based approach is presented. In §4, we describe the datasets and the evaluation metrics used in this study and also the results of this study. In §5, we conclude this paper.

2 Related Work

Riloff et al. (1996) [9] evaluated the relevance of a pattern for IE before application and employed a weighted conditional probability associating higher weight for high frequency to choose the best patterns. Bring (1998) [16] evaluated the derived patterns based on the specificity of the pattern, measuring the length of the middle context, prefix and suffix of the pattern. Patterns with low specificity

were rejected to avoid overly general patterns. Thelen et al. (2002) [17] applied the ranking measure proposed by Riloff et al. (1996) [9] to learn semantic lexicons using extraction pattern contexts. Studies have also evaluated patterns based on their confidence by counting the number of positive and negative entities extracted by the pattern [10–12]. Agichtein et al. (2000) [10] also adopt the ranking measure of Riloff et al. (1996) [9] to consider the coverage of the pattern for evaluation.

Patwardhan et al. (2006) [18] computed "semantic affinity" as a ratio of the target semantic class extractions for each noun class over the total noun class extractions for a closed set of semantic categories. Patwardhan et al. (2007) [19] presented an IE system that decouples the tasks of finding relevant regions of text and applying extraction patterns; a sentence classifier was developped to identify relevant regions and employ semantic affinity measures to automatically learn domain-relevant extraction patterns. Alfonseca et al. (2012) [20] employed topic models to discriminate ambiguous patterns and learn more useful high-precision patterns. Goudong et al. (2005) [21] have shown that diverse lexical, syntactic and semantic knowledge are useful for relation extraction.

Thus, a significant number of studies have investigated various types of patterns, but have generally focused on global estimates such as frequency and accuracy to evaluate patterns. In comparison to the related studies, the focus of the study presented in this paper is not on creating a new type of of pattern for entity extraction. However, unlike studies using global estimates, the goal of the study presented in this paper is to examine the set of available different types of patterns and identify the best patterns to apply in the context of a given sentence. This is achieved, as noted above, by developing a binary classifier that learns from features drawn from sentence-pattern pairings as explained further in the next section.

3 Classification based Pattern Selection

The classification-based method to select a set of patterns per sentence for relation instance extraction is presented in this section. Given a tuple (R, s_j, p_i), consisting of a relation R, $s_j \in \mathcal{S}$ (set of sentences) and a pattern $p_i \in \mathcal{P}$ (set of different types of patterns such as lexical, syntactic etc.), a binary classifier h is trained to return the following prediction for a tuple (R, s_j, p_i):

$$h(R, s_j, p_i) = \begin{cases} +1 & \text{if } p_i \text{ correctly extracts both entities in } s_j \\ -1 & \text{otherwise} \end{cases} \quad (1)$$

Each sentence-pattern pair for a specific relation type is represented as a feature vector $\phi(R, s_j, p_i)$ comprising features {pattern_features, hybrid_features} and labels $\{1, -1\}$ to indicate whether pattern p_i correctly extracts arguments from sentence s_j. While any classifier such as Perceptron, logistic regression or Support Vector Machine (SVM) can be used to construct the above mapping function h, we used in this study an SVM [22] to construct the mapping function h.

3.1 Pattern selection using the classifier

Given a test sentence s_j, the binary classifier selects an optimum set of patterns $\mathcal{P}(s_j)$ such that $\mathcal{P}(s_j) \in s_j = \{p_i : h(p_i, s_j) = 1\}$ i.e., the subset of patterns $\mathcal{P}(s_j) \in s_j$ consists patterns p_i that are classified as true, given the classifier. Further the subset of patterns are ranked based on the confidence scores provided by the classifier. This is achieved by fitting a logistic regression model on top of the distance measure from the decision hyperplane in SVM. LIBSVM [23], a standard library for SVM is used to train the classifer and to fit the logistic regression model to derive confidence scores.

3.2 Features

The different features for the classifier examined in this study are focused on using n-grams from sentences and patterns and other information such as pattern type and length of the pattern. However, the features for the classifier need not be limited to these features alone and can include other features as well. For example, word embeddings can be used as features to represent the sentence-pattern pair. The different features used in this study are described below:

sentence features. The sentence features are designed to capture the context of the sentence and includes the following:

(a) *n-grams in sentence* - the unigram and the bigram terms in the sentence are used as feature terms.

(b) *sentence length* - information about the length of the sentence is provided as features to the classifier. Three features are defined to represent sentence length based on the number of tokens n in the sentence s and includes the following: (a) *sentence_ length_ small* if $n \leq 10$; *sentence_ length_ medium* if $n > 10$ *and* ≤ 30; and *sentence_ length_ long* if $n > 30$.

pattern features. The pattern features are designed based on the information obtained from the patterns and includes the following:

(a) *pattern type* - this feature distinguishes between different types of patterns i.e., whether a given pattern is a lexical or syntactic pattern based on words or grammatical relations or both, and semantic pattern. For example, a feature *pattern_ type_ lexical* is created to indicate a lexical pattern.

(b) *n-grams in pattern* - the unigrams in the pattern are used as feature terms.

(c) *length of patterns* - information about the length of the pattern is also used as a feature. Three features are defined to capture pattern length based on the number of tokens n in pattern p and includes the following: (a) *pattern_ length_ small* if $n \leq 10$; *pattern_ length_ medium* if $n > 10$ *and* ≤ 30; and *pattern_ length_ long* if $n > 30$. In addition to this information, the pattern type is also appended to pattern length to indicate the pattern type. For example, a lexical pattern with less than 10 tokens would be represented using the feature *lexical_pattern_length_small*

(d) *position of patterns* - the position of the pattern in terms of its order of occurrence in the sentence is provided as a feature to the classifier.

3.3 Pattern types

An important aspect of the proposed classification based approach for per sentence pattern selection presented in this paper is the ability for the classifier to select from the available different types of patterns for a given sentence. This study considered three different types of patterns for the classifier (a) lexical patterns; (b) syntactic patterns based on dependency parse; and (c) frame-based semantic patterns. It needs to be noted that the classifier is not confined to these three types of patterns. The classifier can be provided with other types of patterns that can be created for sentences. For example, patterns based on POS tags or named entity recognition can be used with the classifier. The process of deriving the pattern types considered in this study is explained below with the following example sentence:

1. $_{\text{company}}$[Fenrir Inc] is based in the district of $_{\text{location}}$[Osaka], $_{\text{location}}$[Japan].

(a) Lexical Patterns. Following [1], regular expressions are used to define lexical patterns simply comprising lexical entries between the relevant entities as shown in List 1. The arguments for each pattern is shown in parenthesis following the pattern.

List 1 - Lexical patterns: (1) COMPANY is based in the district of LOCATION (Fenrir Inc, Osaka); (2) COMPANY is based in the district of LOCATION, LOCATION (Fenrir Inc, Japan)

(b) Syntactic Patterns. Syntactic patterns for sentences are defined using the shortest path in the dependency graph [2]. In this study, the following three variants of syntactic patterns are used: (a) patterns using words (List 2); (b) patterns using grammatical relations (grs) (List 3); and (c) patterns using both words and grs (List 4). The STANFORD PARSER [24] is used to obtain dependency parse for sentences in this study.

List 2 - Syntactic patterns using words in shortest path: (1) COMPANY is based district LOCATION LOCATION (Fenrir Inc, Osaka); (2) COMPANY is based district LOCATION (Fenrir Inc, Japan)

List 3 - Syntactic patterns using grammatical relations in shortest path: (1) COMPANY nsubj prep prep_in prep_of nn LOCATION (Fenrir Inc, Osaka); (2) COMPANY nsubj prep prep_in prep_of LOCATION (Japan, Osaka)

List 4: Syntactic patterns using words and grammatical relations in shortest path: (1) COMPANY nsubj_is_COMPANY based prep_based_is district prep_in_district prep_of_LOCATION nn_LOCATION_LOCATION LOCATION (Fenrir Inc, Osaka); (2) COMPANY nsubj_is_COMPANY based prep_based_is district prep_in_district prep_of_LOCATION LOCATION (Fenrir Inc, Japan)

(c) Frame-based Semantic Patterns. The study also considers frame-based semantic patterns following the frame semantic framework [25]. Frame

semantics assign *semantic frame elements* to words in a sentence [25] to provide a meaningul represenations for lexical entries in the sentence. Semantic parsing tools such as SEMAFOR [26] is used to derive such semantic frames. The semantic parse obtained using SEMAFOR for the example sentence above is provided below and the semantic patterns obtained using the semantic parse is shown in List 5.

Frame based semantic parse for Sentence 1: $_{\text{Businesses}}$[FENRIR INC] is based in $_{\text{Political\_locales}}$[DISTRICT] of $_{\text{Locale}}$[OSAKA] $_{\text{Locale}}$[JAPAN].

List 5: Frame-based semantic patterns: (1) COMPANY Businesses Political_locales LOCATION LOCATION (Fenrir Inc, Osaka); (2) COMPANY Businesses Political_locales LOCATION (Fenrir Inc, Japan)

4 Experiments

We explain in this section the two datasets used in this study in §4.1. We also describe the evaluation technique employed in this study in §4.2, where we explain the process of deriving negative samples for the classifier, the evaluation metrics and provide more details of the evaluation method followed in this study. Finally, in §4.3, we present the results of obtained in this study.

4.1 Datasets

Wikipedia dataset. The distant supervision method was followed to create the Wikipedia dataset. Specifically, we find all sentences that mentions a pair of entities in the seed dataset, and consider those sentences as describing the semantic relationship between the two entities specified in the seed dataset. DBpedia [27] was used to obtain seed entity pairs for ten different relations, which were further used to obtain sentences from Wikipedia dump. Sentences with a mention of at least one entity pair was retained. The dataset for each relation (details are provided in Table 1) was randomly split in the ratio of 80:20 to create the training and the test set, respectively.

Relation	EP	TS	Relation	EP	TS
ACTOR-MOVIE	1613	3147	COMPANY-FOUNDER	2062	14489
COMPANY-LOCATION	4550	6908	ALBUM-ARTIST	9474	20961
COMPANY-PRODUCT	2890	9122	BIRTHPLACE-PERSON	4114	21737
DIRECTOR-MOVIE	6537	10651	ALBUM-GENRE	8360	22934
AUTHOR-BOOKTITLE	5076	12245	COUNTRY-CITY	4647	45981
Total number of sentences: 168175					

Table 1: Relations in Wikipedia dataset. EP: Entity Pairs; TS: Total Sentences

Riedel et al. (2010) [15] Dataset. The Riedel et al. (2010) [15] dataset was developed with a focus to relax the distant supervision assumption to extract

relations from newswire instead of Wikipedia. However, Freebase was chosen as the knowledge base for obtaining relations and seed entities. Sentences containing two related entities were extracted from the New York Times data, resulting in a large dataset. In this study, we considered ten relations from this dataset (details are provided in Table 2) to evaluate the proposed classifier-based pattern selection method for relation extraction. Sentences for each of these relations were randomly split in the ratio of 80:20 to create the training and test set, respectively.

	Relation	TS
REL_1	people_deceased_person_place_of_death	2541
REL_2	people_person_place_of_birth	4265
REL_3	business_person_company	7987
REL_4	location_administrative_division_country	8860
REL_5	location_country_administrative_divisions	8860
REL_6	location_neighborhood_neighborhood_of	9472
REL_7	people_person_place_lived	9829
REL_8	location_country_capital	11216
REL_9	people_person_nationality	11446
REL_10	location_location_contains	75969
Total number of sentences: 150445		

Table 2: Relations considered from Riedel et al. (2010) [15] dataset; TS: Total Sentences

4.2 Evaluation

Negative Samples for the Classifier. The process of developing the dataset based on the distant supervision method allows to create patterns that extract correct entities. However, the proposed method of developing the classifier for predicting patterns requires negative samples, i.e., patterns that extract wrong entities. The process of deriving negative samples for the classifier is explained below using the following example sentence:

> COMPANYInterwoven was founded in NUM1995 in LOCATIONCalifornia by Peng Tsin Ong of LOCATIONSingapore, who was also COMPANYInterwoven's first CEO and chairman.

In the sentence above, the entity pair (*Interwoven, California*) is the correct argument for the COMPANY-LOCATION relation. Such entity pairs (seed instances) can be obtained from different knowledge sources such as DBpedia and Freebase. The distant supervision method allows us to derive the following two lexical patterns for extract the entities (Interwoven, California). The year information (1995) is changed to NUM and the word Singapore is changed to LOCATION to generalize the patterns.

Patterns extracting correct entities

1. <COMPANY> was founded in NUM in <LOCATION>
2. <LOCATION> by Peng Tsin Ong of LOCATION, who was also <COMPANY>

However, the presence of the LOCATION entity term 'Singapore' allows to derive the following lexical patterns that extract wrong entities (*Interwoven, Singapore*) for the COMPANY-LOCATION relation:

Patterns extracting wrong entities

1. <COMPANY> was founded in NUM in LOCATION by Peng Tsin Ong of <LOCATION>
2. <LOCATION>, who was also <COMPANY>

Thus, the patterns extracting wrong entities are used as negative samples for the classifier.

Evaluation Metrics. Further, given a pattern $l \in \mathcal{L}$, the pattern set obtained from train data, and a test sentence $s \in \mathcal{S}$, the following types of patterns are defined:

1. *matched pattern*: the pattern l is defined as a *matched pattern* for the test sentence s, iff (if and only if) the pattern l matches the test sentence s.
2. *correct pattern*: a pattern l is defined as a *correct pattern* for the test sentence s, iff the pattern l matches the test sentence s and correctly extracts the two arguments (e_1, e_2) for a given relation r.

The precision of a pattern l is defined as the ratio of number of times the pattern l is seen as a *correct pattern* to the number of times it is seen as a matched pattern on the test set $\mathcal{S}$. Thus, the precision of a pattern l on the test set $\mathcal{S}$ is given by:

$$\text{Precision } (l) = \frac{\#\text{ pattern } l \text{ is a } \textit{correct pattern} \text{ in } \mathcal{S}}{\#\text{ pattern } l \text{ is a } \textit{matched pattern} \text{ in } \mathcal{S}} \tag{2}$$

The overall precision P of the pattern set is obtained by:

$$P = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \text{Precision } (l) \tag{3}$$

where $|\mathcal{L}|$ is the total number of patterns in the pattern set.

The recall of a pattern set is measured in terms of its effectiveness or coverage in applying correct patterns on the test set and is defined as the ratio of of the total number of test sentences on which *correct patterns* are applied to the total number of test sentences. Thus, the recall of a pattern set is given by:

$$R = \frac{\#\text{ of test sentences with } \textit{correct patterns}}{\#\text{ of test sentences}} \tag{4}$$

Given Precision P and Recall R, the F-score of a pattern set is obtained by:

$$\text{F-Score} = \frac{2 \times PR}{P + R} \tag{5}$$

The classification accuracy of the classifier on the test set is reported on two different set of features: (a) pattern features; (b) hybrid features - combining features obtained from sentences and patterns. If p_c is the total number of correctly classified patterns and p_t is the total number of patterns in test set, the accuracy of the classifier Accuracy(c) is obtained using the equation:

$$\text{Accuracy(c)} = \frac{p_c}{p_t} \tag{6}$$

Model selection for optimal parameter estimation was performed as a grid search through cross-validation on the development set [28].

Evaluation Method. In a regular setting, the large pattern set obtained from the training set are applied on the test set for relation extraction. At this stage, the patterns from the train set that match the patterns in the test set are applied for relation extraction. However, there could be many patterns in the matched pattern set that extract wrong entities for the targeted relation. The proposed per-sentence classifier approach for relation extraction is employed to filter such patterns from the matched pattern set that extract wrong entities. This filtering process can help in achieving higher precision, without losing on recall. Thus, in this study, the matched pattern set from the training set is evaluated against the filtered pattern set obtained using the classifier for relation extraction. The evaluation method employed in this study is shown in Figure 1.

As shown in Figure 1, the following three pattern sets based on their size are evaluated: (a) 10% of most frequent patterns; (b) 50% of most frequent patterns; and (c) full pattern set obtained from the train set. Further, as seen in Figure 1, with regard to applying different sets of frequent patterns, the patterns are initially matched against the test set, following which the *matched patterns* are applied to identify *correct patterns*. However, with regard to the classifier, the *matched pattern* set is examined against the classifier to obtain a *filtered pattern* set, comprising only those patterns that are positively classified by the classifier. The *filtered pattern* is now applied on the test set for relation extraction. The precision, recall and F-score for the applied pattern sets in both cases are recorded for different relations.

4.3 Results

The evaluation results of applying the classifier approach proposed in this study against the regular use of frequency based pattern set for relation extraction is presented in this section.

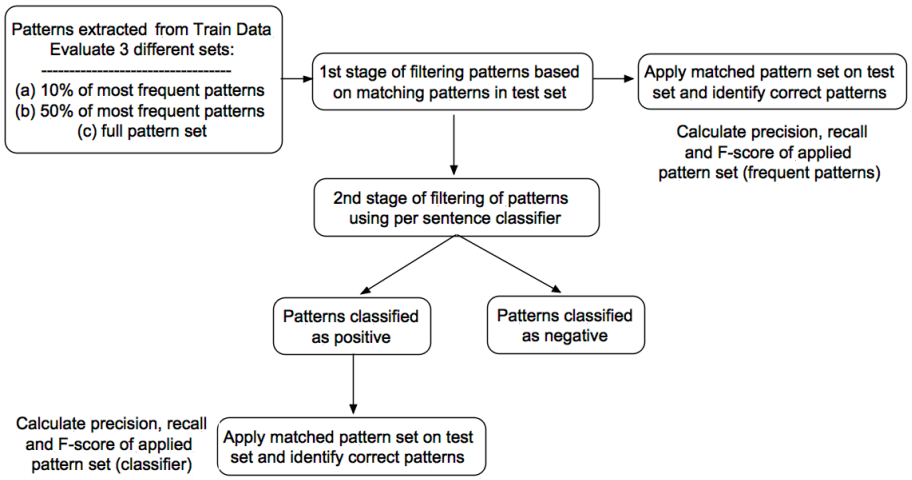

Fig. 1: Evaluation methodology for evaluating pattern set obtained using frequent patterns and classifier.

Classifier vs. high frequency patterns As seen in Tables 3 and 4, the per sentence classifier-based approach for relation extraction, achieves statistically significant average F-scores ($p \leq 0.05$; Wilcoxon Signed-Rank Test) for relations both in Wikipedia dataset and Riedel et al. (2010) [15] dataset for pattern sets varying in different sizes. For example, for relations in Wikipedia dataset (Table 3) the classifier achieves statistically significant average F-scores of 0.81, 0.83 and 0.85 ($p \leq 0.05$; Wilcoxon Signed-Rank Test) against the average F-score of 0.72, 0.73 and 0.70 achieved by using 10% and 50% of high frequency patterns, and the full pattern set obtained from the training set, respectively. A similar performance is seen for relations in Riedel et al. (2010) [15] dataset (Table 4), with the classifier achieving statistically significant average F-scores of 0.76, 0.84 and 0.88 ($p \leq 0.05$; Wilcoxon Signed-Rank Test) against the average F-score values of 0.69, 0.78 and 0.80 achieved for 10% and 50% of high frequency patterns, and the full pattern set obtained from the training set, respectively. These results indicate that classifier-based approach is significantly useful for the task of relation extraction.

Further as seen in Tables 3 and 4, the increase in the size of the applied pattern set on test sentence, results in a significant increase in the performance of the classifier. For example, with the Wikipedia dataset, while the classifier achieves an average F-score of 0.81 with 10% of high frequency patterns, the classifier achieves a statistically significant higher average F-score of 0.83 using 50% of high frequency patterns ($p \leq 0.05$; Wilcoxon Signed-Rank Test). The classifier achieves a further higher average F-score of 0.85 with the use of full-pattern set obtained from the training data. A similar performance is also seen for relations in Riedel et al. (2010) [15] dataset as shown in Table 4. While the classifier

Relation	10% patterns		50% patterns		Full pattern set	
	Frequent Patterns	Per Sentence Classifier	Frequent Patterns	Per Sentence Classifier	Pattern Set	Per Sentence Classifier
ACTOR-MOVIE	0.72	**0.79**	0.73	**0.82**	0.74	**0.84**
COMPANY-LOCATION	0.72	**0.80**	0.73	**0.82**	0.73	**0.84**
COMPANY-PRODUCT	0.76	**0.82**	0.77	**0.83**	0.77	**0.84**
DIRECTOR-MOVIE	0.70	**0.84**	0.70	**0.86**	0.68	**0.87**
AUTHOR-BOOKTITLE	0.75	**0.83**	0.74	**0.84**	0.74	**0.85**
COMPANY-FOUNDER	0.82	**0.85**	0.83	**0.87**	0.84	**0.88**
ALBUM-ARTIST	0.69	**0.80**	0.69	**0.83**	0.69	**0.84**
BIRTHPLACE-PERSON	0.64	**0.75**	0.64	**0.78**	0.64	**0.80**
ALBUM-GENRE	0.69	**0.81**	0.69	**0.83**	0.69	**0.85**
COUNTRY-CITY	0.75	**0.82**	0.76	**0.85**	0.77	**0.86**
Average	0.72	**0.81***	0.73	**0.83***†	0.70	**0.85***†

Table 3: F-score values for relations in Wikipedia dataset. *statistically significant against applying patterns based on frequency and full pattern set. †statistically significant than using the previous pattern set size.

Relation	10% patterns		50% patterns		Full pattern set	
	Patterns Only	Classifier	Patterns Only	Classifier	Patterns Only	Classifier
REL_1	0.57	**0.57**	0.65	**0.66**	0.68	**0.69**
REL_2	0.64	**0.68**	0.73	**0.7**9	0.79	**0.90**
REL_3	0.85	**0.87**	0.86	**0.91**	0.84	**0.92**
REL_4	0.70	**0.72**	0.79	**0.84**	0.82	**0.89**
REL_5	0.71	**0.73**	0.79	**0.83**	0.82	**0.87**
REL_6	0.81	**0.87**	0.84	**0.92**	0.83	**0.94**
REL_7	0.74	**0.80**	0.78	**0.86**	0.80	**0.89**
REL_8	0.68	**0.71**	0.75	**0.82**	0.81	**0.90**
REL_9	0.60	**0.85**	0.80	**0.89**	0.79	**0.90**
REL_10	0.63	**0.88**	0.82	**0.91**	0.84	**0.92**
Average	0.69	**0.76***	0.78†	**0.84***†	0.80†	**0.88***†

Table 4: F-score values for relations in Riedel et al. (2010) [15] dataset. *statistically significant against applying patterns based on frequency and full pattern set. †statistically significant than using the previous pattern set size.

achieves an average F-scores of 0.76 with the use of 10% of high frequency patterns, a statistically significant higher average F-score o 0.84 is achieved using 50% of high frequency patterns ($p \leq 0.05$; Wilcoxon Signed-Rank Test. A higher performance is achieved with the use of full pattern set, with the classifier achieving a higher average F-score of 0.88.

The precision values scored for relations in Wikipedia dataset and Riedel et al. (2010) [15] dataset is shown in Figures 2 and 3. The recall values are not reported here, since both the classifier and high frequency patterns achieve the

same recall for all relations for both the datasets. As seen in Figures 2 and 3, the increase in the applied pattern set size results in a decrease in precision values for majority of relations in both the datasets. This can be the reason for the decrease in the performance of high frequency patterns, when larger set of patterns are used. However, on the other hand, the precision score improves with the increase in the applied pattern set size for all relations. These results further prove the usefulness of classifier-based approach for the purpose of relation extraction.

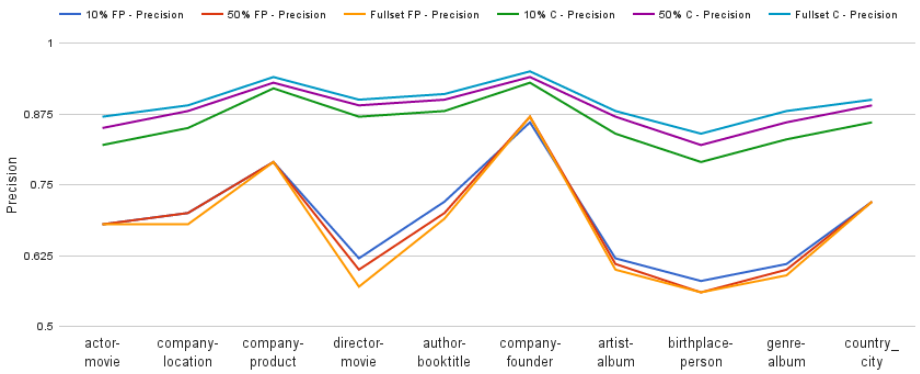

Fig. 2: Precision values for relations in Wikipedia dataset.

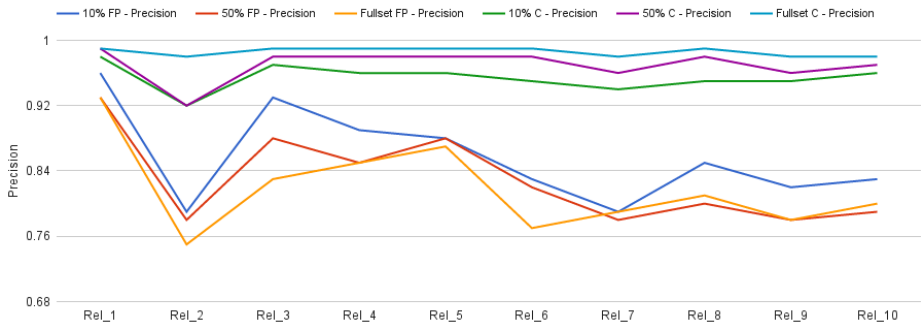

Fig. 3: Precision values for relations in Riedel et al. (2010) [15] dataset.

Wikipedia vs. Riedel et al. (2010) [15] dataset The proposed classifier-based pattern selection approach was evaluated on relations drawn from two dif-

ferent datasets: (a) Wikipedia dataset; and (b) Riedel et al. (2010) [15] dataset. Interestingly, the increase in the size of high frequency patterns lowers the performance of relation extraction, particularly for the Wikipedia dataset. As seen in Table 3, while an average F-score of 0.72 is achieved with 10% of higher frequency patterns, a slightly higher F-score of 0.73 is achieved with using 50% of high frequency patterns. However, with the use of the full pattern set obtained from the training data, a further lower average F-score of 0.70 is achieved. The decrease in the performance with the increase in the size of the pattern set is not statistically significant. This indicates that a significantly large proportion of patterns seen in the test sentences are covered in the top 10% of high frequency patterns, indicating the usefulness of higher frequency patterns.

However in the case of Riedel et al. (2010) [15] dataset, the increase in the pattern set size of high frequency patterns does not lower the performance of relation extraction. The performance obtained for larger pattern set of high frequency patterns is statistically significant, with an average F-score of 0.78 being achieved with 50% of high frequency patterns, while a lower average F-score of 0.69 is obtained with 10% of higher frequency patterns ($p \leq 0.05$; Wilcoxon Signed-Rank Test). The average F-score (0.80) obtained using the full pattern set is further higher than employing smaller proportions of high frequency patterns.

These results show that using high frequency patterns and the full pattern set for relation extraction is more beneficial for Riedel et al. (2010) dataset. It needs to be noted that the Wikipedia dataset is developed based on the distant supervision method where training knowledge base derived from the training text is employed to obtain patterns from the training text. However, the Riedel et al. dataset was developed by relaxing the distant supervision assumption, where an external training knowledge base, not derived from the training text is used for pattern extraction. Thus, in the case of Wikipedia dataset, the obtained pattern set from training data suffers from poor precision i.e., extract wrong entities in spite of matching the test sentences, resulting in the poor performance for relation extraction. However, the patterns obtained in Riedel et al. (2010) dataset are more precise, extracting correct entities given match on test sentences. These results show that high frequency patterns are more useful for datasets where distant supervision assumption is relaxed. However, the classifier-based pattern selection approach surpasses the performance obtained using high frequency patterns and the complete pattern set, indicating the usefulness of classifier-based approach on both types of datasets.

Classification Accuracy of the Classifier As mentioned previously in §3 SVM was adopted as a binary classifier for this study. The following two types of feature sets were examined: (a) pattern features - features obtained from pattern alone; and (b) hybrid features - combining pattern features along with features obtained from the sentence. The pattern and sentence features were previously discussed in §3. To choose the best kernel for SVM, the classification accuracy of the classifier for various kernels was examined for the development set of COMPANY-LOCATION as shown in Table 5. As seen in Table 5, the Radial

Basis Function (RBF) kernel using the optimal parameters from grid search, achieved the best performance scoring an accuracy of 69.35 for hybrid features (HF).

Further, the performance of various kernels was also examined for the task of relation extraction for the COMPANY-LOCATION relation involving different sets of features as shown in Table 5. As seen in Table 5, the RBF kernel achieved the best performance using hybrid feaures (HF), scoring the best F-score of 0.80, 0.82, and 0.84 for 10% and 50% of high frequency patterns, and the full pattern set obtained from the training set, respectively. Based on these results on the development set, the RBF kernel was chosen to examine the classifier accuracy on the remaining nine datasets.

Function	PF	HF
Linear	63.42	63.85
Polynomial (degree 2)	64.71	68.96
Polynomial (degree 3)	62.08	67.05
Radial Basis	65.21	69.35

Table 5: Classification accuracy (c) of the classifier for various kernels for COMPANY-LOCATION dataset (development set). PF - pattern features, HF - hybrid features

Wikipedia Dataset			Riedel et al. (2010) Dataset [15]		
Relation	PF	HF	Relation	PF	HF
ACTOR-MOVIE	67.24	68.57	REL_1	70.79	88.81
COMPANY-LOCATION	65.21	69.35	REL_2	74.67	79.55
COMPANY-PRODUCT	68.62	75.55	REL_3	80.44	88.74
DIRECTOR-MOVIE	76.20	77.04	REL_4	75.44	87.26
AUTHOR-BOOKTITLE	74.24	74.34	REL_5	76.11	87.65
COMPANY-FOUNDER	68.48	82.68	REL_6	82.15	90.84
ALBUM-ARTIST	69.33	71.41	REL_7	79.61	86.17
BIRTHPLACE-PERSON	60.84	61.52	REL_8	72.95	81.60
ALBUM-GENRE	67.68	71.21	REL_9	74.51	82.66
COUNTRY-CITY	64.06	71.14	REL_10	73.45	81.45
AVERAGE	68.19	**72.28***		76.59	**85.10***

Table 6: Classification accuracy of the classifier on different datasets. PF - pattern features, HF - hybrid features, *performance obtained using hybrid features is statistically significant than using pattern features alone.

5 Conclusion

We presented in this paper a classifier-based pattern selection approach for relation instance extraction. The classifier-based approach for relation extraction was evaluated against using different proportions of high frequency patterns and also employing the full pattern set for relation extraction. This paper showed that employing a classifier to remove patterns that extract wrong entities for a given relation before applying on test sentences, helps in improving precision without compromising on recall, which in turn facilitate significant improvement in the relation extraction task. The results show that an increase in the applied high frequency patterns results in lowering the performance for relation extraction, particularly on datasets developed based on distant supervision method. The results further show that the classifier-based pattern selection approach is useful for relation extraction on different types of datasets that are developed following distant supervision and also where the distant supervision assumption is relaxed.

References

1. Ravichandran, D., Hovy, E.: Learning surface text patterns for a question answering system. In: Proceedings of the COLING, Association for Computational Linguistics (2002) 41–47
2. Wu, F., Weld, D.S.: Open information extraction using wikipedia. In: Proceedings of COLING, Association for Computational Linguistics (2010) 118–127
3. Etzioni, O., Fader, A., Christensen, J., Soderland, S., Mausam, M.: Open information extraction: The second generation. In: Proceedings of IJCAI. Volume 11. (2011) 3–10
4. Gamallo, P., Garcia, M., Fernández-Lanza, S.: Dependency-based open information extraction. In: Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP, Association for Computational Linguistics (2012) 10–18
5. Kim, J.T., Moldovan, D., et al.: Acquisition of linguistic patterns for knowledge-based information extraction. Knowledge and Data Engineering, IEEE Transactions on **7** (1995) 713–724
6. Moschitti, A., Morarescu, P., Harabagiu, S.M.: Open domain information extraction via automatic semantic labeling. In: Proceedings of FLAIRS. (2003) 397–401
7. Shen, D., Lapata, M.: Using semantic roles to improve question answering. In: Proceedings of EMNLP-CoNLL. (2007) 12–21
8. Søgaard, A., Plank, B., Alonso, H.M.: Using frame semantics for knowledge extraction from twitter. In: Proceedings of AAAI. (2015)
9. Riloff, E.: Automatically generating extraction patterns from untagged text. In: Proceedings of the national conference on artificial intelligence. (1996) 1044–1049
10. Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: Proceedings of the fifth ACM conference on Digital libraries, ACM (2000) 85–94
11. Yangarber, R., Lin, W., Grishman, R.: Unsupervised learning of generalized names. In: Proceedings of COLING, Association for Computational Linguistics (2002) 1–7

12. Lin, W., Yangarber, R., Grishman, R.: Bootstrapped learning of semantic classes from positive and negative examples. In: Proceedings of ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data. Volume 1. (2003) 21
13. Gupta, S., Manning, C.D.: Improved pattern learning for bootstrapped entity extraction. In: CoNLL. (2014) 98–108
14. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, Association for Computational Linguistics (2009) 1003–1011
15. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer (2010) 148–163
16. Brin, S.: Extracting patterns and relations from the world wide web. In: The World Wide Web and Databases. Springer (1998) 172–183
17. Thelen, M., Riloff, E.: A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In: Proceedings of EMNLP, Association for Computational Linguistics (2002) 214–221
18. Patwardhan, S., Riloff, E.: Learning domain-specific information extraction patterns from the web. In: Proceedings of the Workshop on Information Extraction beyond the Document, Association for Computational Linguistics (2006) 66–73
19. Patwardhan, S., Riloff, E.: Effective information extraction with semantic affinity patterns and relevant regions. In: Proceedings of EMNLP-CoNLL. Volume 7. (2007) 717–727
20. Alfonseca, E., Filippova, K., Delort, J.Y., Garrido, G.: Pattern learning for relation extraction with a hierarchical topic model. In: Proceedings of the 50th Annual Meeting of the ACL, Association for Computational Linguistics (2012) 54–59
21. GuoDong, Z., Jian, S., Jie, Z., Min, Z.: Exploring various knowledge in relation extraction. In: Proceedings of COLING, Association for Computational Linguistics (2005) 427–434
22. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20** (1995) 273–297
23. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (2011) 27:1–27:27
24. De Marneffe, M.C., MacCartney, B., Manning, C.D., et al.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC. Volume 6. (2006) 449–454
25. Fillmore, C.: Frame semantics. Linguistics in the morning calm (1982) 111–137
26. Das, D., Chen, D., Martins, A.F., Schneider, N., Smith, N.A.: Frame-semantic parsing. Computational Linguistics **40** (2014) 9–56
27. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: In 6th Intl Semantic Web Conference, Busan, Korea. (2007)
28. Hsu, C.W., Chang, C.C., Lin, C.J., et al.: A practical guide to support vector classification. (2003)