Relation Extraction with Relation Topics

Chang Wang **James Fan** **Aditya Kalyanpur** **David Gondek**
IBM T. J. Watson Research Lab
19 Skyline Drive, Hawthorne, New York 10532
{wangchan, fanj, adityakal, dgondek}@us.ibm.com

Abstract

This paper describes a novel approach to the semantic relation detection problem. Instead of relying only on the training instances for a new relation, we leverage the knowledge learned from previously trained relation detectors. Specifically, we detect a new semantic relation by projecting the new relation's training instances onto a lower dimension topic space constructed from existing relation detectors through a three step process. First, we construct a large relation repository of more than 7,000 relations from Wikipedia. Second, we construct a set of non-redundant relation topics defined at multiple scales from the relation repository to characterize the existing relations. Similar to the topics defined over words, each relation topic is an interpretable multinomial distribution over the existing relations. Third, we integrate the relation topics in a kernel function, and use it together with SVM to construct detectors for new relations. The experimental results on Wikipedia and ACE data have confirmed that background-knowledge-based topics generated from the Wikipedia relation repository can significantly improve the performance over the state-of-the-art relation detection approaches.

1 Introduction

Detecting semantic relations in text is very useful in both information retrieval and question answering because it enables knowledge bases to be leveraged to score passages and retrieve candidate answers. To extract semantic relations from text, three types of approaches have been applied. Rule-based methods (Miller et al., 2000) employ a number of linguistic rules to capture relation patterns. Feature-based methods (Kambhatla, 2004; Zhao and Grishman, 2005) transform relation instances into a large amount of linguistic features like lexical, syntactic and semantic features, and capture the similarity between these feature vectors. Recent results mainly rely on kernel-based approaches. Many of them focus on using tree kernels to learn parse tree structure related features (Collins and Duffy, 2001; Culotta and Sorensen, 2004; Bunescu and Mooney, 2005). Other researchers study how different approaches can be combined to improve the extraction performance. For example, by combining tree kernels and convolution string kernels, (Zhang et al., 2006) achieved the state of the art performance on ACE (ACE, 2004), which is a benchmark dataset for relation extraction.

Although a large set of relations have been identified, adapting the knowledge extracted from these relations for new semantic relations is still a challenging task. Most of the work on domain adaptation of relation detection has focused on how to create detectors from ground up with as little training data as possible through techniques such as bootstrapping (Etzioni et al., 2005). We take a different approach, focusing on how the knowledge extracted from the existing relations can be reused to help build detectors for new relations. We believe by reusing knowledge one can build a more cost effective relation detector, but there are several challenges associated with reusing knowledge.

The first challenge to address in this approach is how to construct a relation repository that has suffi-

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1426–1436,
Edinburgh, Scotland, UK, July 27–31, 2011.

cient coverage. In this paper, we introduce a method that automatically extracts the knowledge characterizing more than 7,000 relations from Wikipedia. Wikipedia is comprehensive, containing a diverse body of content with significant depth and grows rapidly. Wikipedia's infoboxes are particularly interesting for relation extraction. They are short, manually-created, and often have a relational summary of an article: a set of attribute/value pairs describing the article's subject.

Another challenge is how to deal with overlap of relations in the repository. For example, Wikipedia authors may make up a name when a new relation is needed without checking if a similar relation has already been created. This leads to relation duplication. We refine the relation repository based on an unsupervised multiscale analysis of the correlations between existing relations. This method is parameter free, and able to produce a set of non-redundant *relation topics* defined at multiple scales. Similar to the topics defined over words (Blei et al., 2003), we define relation topics as multinomial distributions over the existing relations. The relation topics extracted in our approach are interpretable, orthonormal to each other, and can be used as basis relations to re-represent the new relation instances.

The third challenge is how to use the relation topics for a relation detector. We map relation instances in the new domains to the relation topic space, resulting in a set of new features characterizing the relationship between the relation instances and existing relations. By doing so, background knowledge from the existing relations can be introduced into the new relations, which overcomes the limitations of the existing approaches when the training data is not sufficient. Our work fits in to a class of relation extraction research based on "distant supervision", which studies how knowledge and resources external to the target domain can be used to improve relation extraction. (Mintz et al., 2009; Jiang, 2009; Chan and Roth, 2010). One distinction between our approach and other existing approaches is that we represent the knowledge from distant supervision using automatically constructed topics. When we test on new instances, we do not need to search against the knowledge base. In addition, our topics also model the indirect relationship between relations. Such information cannot be directly found from the knowledge base.

The contributions of this paper are three-fold. Firstly, we extract a large amount of training data for more than 7,000 semantic relations from Wikipedia (Wikipedia, 2011) and DBpedia (Auer et al., 2007). A key part of this step is how we handle noisy data with little human effort. Secondly, we present an unsupervised way to construct a set of *relation topics* at multiple scales. This step is parameter free, and results in a non-redundant, multiscale relation topic space. Thirdly, we design a new kernel for relation detection by integrating the relation topics into the relation detector construction. The experimental results on Wikipedia and ACE data (ACE, 2004) have confirmed that background-knowledge-based features generated from the Wikipedia relation repository can significantly improve the performance over the state-of-the-art relation detection approaches.

2 Extracting Relations from Wikipedia

Our training data is from two parts: relation instances from DBpedia (extracted from Wikipedia infoboxes), and sentences describing the relations from the corresponding Wikipedia pages.

2.1 Collecting the Training Data

Since our relations correspond to Wikipedia infobox properties, we use an approach similar to that described in (Hoffmann et al., 2010) to collect positive training data instances. We assume that a Wikipedia page containing a particular infobox property is likely to express the same relation in the text of the page. We further assume that the relation is most likely expressed in the first sentence on the page which mentions the arguments of the relation. For example, the Wikipedia page for "Albert Einstein" contains an infobox property "alma mater" with value "University of Zurich", and the first sentence mentioning the arguments is the following: "Einstein was awarded a PhD by the University of Zurich", which expresses the relation. When looking for relation arguments on the page, we go beyond (sub)string matching, and use link information to match entities which may have different surface forms. Using this technique, we are able to collect a large amount of positive training instances of DBpe-

dia relations.

To get precise type information for the arguments of a DBpedia relation, we use the DBpedia knowledge base (Auer et al., 2007) and the associated YAGO type system (Suchanek et al., 2007). Note that for every Wikipedia page, there is a corresponding DBpedia entry which has captured the infobox-properties as RDF triples. Some of the triples include type information, where the subject of the triple is a Wikipedia entity, and the object is a YAGO type for the entity. For example, the DBpedia entry for the entity "Albert Einstein" includes YAGO types such as Scientist, Philosopher, Violinist etc. These YAGO types are also linked to appropriate WordNet concepts, providing for accurate sense disambiguation. Thus, for any entity argument of a relation we are learning, we obtain sense-disambiguated type information (including super-types, sub-types, siblings etc.), which become useful generalization features in the relation detection model. Given a common noun, we can also retrieve its type information by checking against WordNet (Fellbaum, 1998).

2.2 Extracting Rules from the Training Data

We use a set of rules together with their popularities (occurrence count) to characterize a relation. A rule representing the relations between two arguments has five components (ordered): argument_1 type, argument_2 type, noun, preposition and verb. A rule example of *ActiveYearsEndDate* relation (about the year that a person retired) is:

person100007846|year115203791|-|in|retire.

In this example, argument_1 type is *person100007846*, argument_2 type is *year115203791*, both of which are from YAGO type system. The key words connecting these two arguments are *in* (preposition) and *retire* (verb). This rule does not have a noun, so we use a '-' to take the position of noun. The same relation can be represented in many different ways. Another rule example characterizing the same relation is

person100007846|year115203791|retirement|-|announce.

This paper only considers three types of words: *noun*, *verb* and *preposition*. It is straightforward to expand or simplify the rules by including more or removing some word types. The keywords are extracted from the shortest path on the dependency tree between the two arguments. A dependency tree (Figure 1) represents grammatical relations between words in a sentence. We used a slot grammar parser (McCord, 1995) to generate the parse tree of each sentence. Note that there could be multiple paths between two arguments in the tree. We only take the shortest path into consideration. The popularity value corresponding to each rule represents how many times this rule applies to the given relation in the given data. Multiple rules can be constructed from one relation instance, if multiple argument types are associated with the instance, or multiple nouns, prepositions or verbs are in the dependency path.

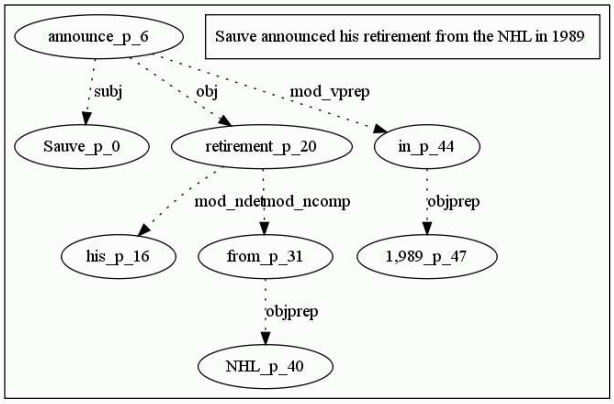

Figure 1: A dependency tree example.

2.3 Cleaning the Training Data

To find a sentence on the Wikipedia page that is likely to express a relation in its infobox, we consider the first sentence on the page that mentions both arguments of the relation. This heuristic approach returns reasonably good results, but brings in about 20% noise in the form of false positives, which is a concern when building an accurate statistical relation detector. To address this issue, we have developed a two-step technique to automatically remove some of the noisy data. In the first step, we extract popular argument types and keywords for each DBpedia relation from the given data, and then use the combinations of those types and words to create initial rules. Many of the argument types and keywords introduced by the noisy data are often not very popular, so they can be filtered out in the first step. Not all initial rules make sense. In the second step, we

check each rule against the training data to see if that rule really exists in the training data or not. If it does not exist, we filter it out. If a sentence does not have a single rule passing the above procedure, that sentence will be removed. Using the above techniques, we collect examples characterizing 7,628 DBpedia relations.

3 Learning Multiscale Relation Topics

An extra step extracting knowledge from the raw data is needed for two reasons: Firstly, many DBpedia relations are inter-related. For example, some DBpedia relations have a subclass relationship, e.g. "AcademyAward" and "Award"; others overlap in their scope and use, e.g., "Composer" and "Artist"; while some are equivalent, e.g., "DateOfBirth" and "BirthDate". Secondly, a fairly large amount of the noisy labels are still in the training data.

To reveal the intrinsic structure of the current DBpedia relation space and filter out noise, we carried out a correlation analysis of relations in the training data, resulting in a relation topic space. Each relation topic is a multinomial distribution over the existing relations. We adapted diffusion wavelets (Coifman and Maggioni, 2006) for this task. Compared to the other well-known topic extraction methods like LDA (Blei et al., 2003) and LSI (Deerwester et al., 1990), diffusion wavelets can efficiently extract a hierarchy of interpretable topics without any user input parameter (Wang and Mahadevan, 2009).

3.1 An Overview of Diffusion Wavelets

The diffusion wavelets algorithm constructs a compressed representation of the dyadic powers of a square matrix by representing the associated matrices at each scale not in terms of the original (unit vector) basis, but rather using a set of custom generated bases (Coifman and Maggioni, 2006). Figure 2 summarizes the procedure to generate diffusion wavelets. Given a matrix T, the $\mathcal{QR}$ (a modified QR decomposition) subroutine decomposes T into an orthogonal matrix Q and a triangular matrix R such that $T \approx QR$, where $|T_{i,k} - (QR)_{i,k}| < \varepsilon$ for any i and k. Columns in Q are orthonormal basis functions spanning the column space of T at the finest scale. RQ is the new representation of T with respect to the space spanned by the columns of Q (this result is based on the matrix invariant subspace theory). At an arbitrary level j, $\mathcal{DWT}$ learns the basis functions from T^{2^j} using $\mathcal{QR}$. Compared to the number of basis functions spanning T^{2^j}'s original column space, we usually get fewer basis functions, since some high frequency information (corresponding to the "noise" at that level) can be filtered out. $\mathcal{DWT}$ then computes $T^{2^{j+1}}$ using the low frequency representation of T^{2^j} and the procedure repeats.

$\{[\phi_j]_{\phi_0}\} = \mathcal{DWT}(T, \varepsilon, J)$
//**INPUT**:
//T: The input matrix.
//ε: Desired precision, which can be set to a small number or simply machine precision.
//J: Number of levels (optional).
//**OUTPUT**:
//$[\phi_j]_{\phi_0}$: extended diffusion scaling functions at scale j.

$\phi_0 = I$;
$For\ j = 0\ to\ J-1\ \{$
  $([\phi_{j+1}]_{\phi_j}, [T^{2^j}]_{\phi_j}^{\phi_{j+1}}) \leftarrow \mathcal{QR}([T^{2^j}]_{\phi_j}^{\phi_j}, \varepsilon)$;
  $[\phi_{j+1}]_{\phi_0} = [\phi_{j+1}]_{\phi_j}[\phi_j]_{\phi_0}$;
  $[T^{2^{j+1}}]_{\phi_{j+1}}^{\phi_{j+1}} = ([T^{2^j}]_{\phi_j}^{\phi_{j+1}}[\phi_{j+1}]_{\phi_j})^2$;
$\}$

Figure 2: Diffusion Wavelets construct multiscale representations of the input matrix at different scales. $\mathcal{QR}$ is a modified QR decomposition. J is the max step number (this is optional, since the algorithm automatically terminates when it reaches a matrix of size 1×1). The notation $[T]_{\phi_a}^{\phi_b}$ denotes matrix T whose column space is represented using basis ϕ_b at scale b, and row space is represented using basis ϕ_a at scale a. The notation $[\phi_b]_{\phi_a}$ denotes basis ϕ_b represented on the basis ϕ_a. At an arbitrary scale j, we have p_j basis functions, and length of each function is l_j. The number of p_j is determined by the intrinsic structure of the given dataset in $\mathcal{QR}$ routine. $[T]_{\phi_a}^{\phi_b}$ is a $p_b \times l_a$ matrix, and $[\phi_b]_{\phi_a}$ is an $l_a \times p_b$ matrix.

3.2 Constructing Multiscale Relation Topics

Learning Relation Correlations

Assume we have M relations, and the i^{th} of them is characterized by m_i <*rule, popularity*> pairs. We use $s(a, b)$ to represent the similarity between the a^{th} and b^{th} relations. To compute $s(a, b)$, we first normalize the popularities for each relation, and then

look for the rules that are shared by both relation a and b. We use the product of corresponding popularity values to represent the similarity score between two relations with respect to each common rule. $s(a, b)$ is set to the sum of such scores over all common rules. The relation-relation correlation matrix S is constructed as follows:

$$S = \begin{bmatrix} s(1,1) & \cdots & s(1,M) \\ \cdots & \cdots & \cdots \\ s(M,1) & \cdots & s(M,M) \end{bmatrix}$$

We have more than $200,000$ argument types, tens of thousands of distinct nouns, prepositions, and verbs, so we potentially have trillions of distinct rules. One rule may appear in multiple relations. The more rules two relations share, the more related two relations should be. The rules shared across different relations offer us a novel way to model the correlations between different relations, and further allow us to create relation topics. The rules can also be simplified. For example, we may treat argument$_1$, argument$_2$, noun, preposition and verb separately. This results in simple rules that only involve in one argument type or word. The correlations between relations are then computed only based on one particular component like argument$_1$, noun, etc.

Theoretical Analysis

Matrix S models the correlations between relations in the training data. Once S is constructed, we adapt diffusion wavelets (Coifman and Maggioni, 2006) to automatically extract the basis functions spanning the original column space of S at multiple scales. The key strength of the approach is that it is data-driven, largely parameter-free and can automatically determine the number of levels of the topical hierarchy, as well as the topics at each level. However, to apply diffusion wavelets to S, we first need to show that S is a positive semi-definite matrix. This property guarantees that all eigenvalues of S are ≥ 0. Depending on the way we formalize the rules, the methods to validate this property are slightly different. When we treat argument$_1$, argument$_2$, noun, preposition and verb separately, it is straightforward to see the property holds. In Theorem 1, we show the property also holds when we use more complicated rules (using the 5-tuple rule in Section 2.2 as an example in the proof).

Theorem 1. *S is a Positive Semi-Definite matrix.*

Proof: An arbitrary rule r_i is uniquely characterized by a five tuple: *argument$_1$ type*| *argument$_2$ type*| *noun*| *preposition*| *verb*. Since the number of distinct argument types and words are constants, the number of all possible rules is also a constant: R.

If we treat each rule as a feature, then the set of rules characterizing an arbitrary relation r_i can be represented as a point $[p_i^1, \cdots, p_i^R]$ in a latent R dimensional rule space, where p_i^j represents the popularity of rule j in relation r_i in the given data.

We can verify that the way to compute $s(a, b)$ is the same as $s(a, b) =< [p_a^1 \cdots p_a^R], [p_b^1 \cdots p_b^R] >$, where $< \cdot, \cdot >$ is the cosine similarity (kernel). It follows directly from the definition of positive semi-definite matrix (PSD) that S is PSD (Schölkopf and Smola, 2002). □

In our approach, we construct multiscale relation topics by applying $\mathcal{DWT}$ to decompose $S/\lambda_{max}(S)$, where $\lambda_{max}(S)$ represents the largest eigenvalue of S. Theorem 2 shows that this decomposition will converge, resulting in a relation topic hierarchy with one single topic at the top level.

Theorem 2. *Let $\lambda_{max}(S)$ represent the largest eigenvalue of matrix S, then $\mathcal{DWT}(S/\lambda_{max}(S), \varepsilon)$ produces a set of nested subspaces of the column space of S, and the highest level of the resulting subspace hierarchy is spanned by one basis function.*

Proof: From Theorem 1, we know that S is a PSD matrix. This means $\lambda_{max}(S) \in [0, +\infty)$ (all eigenvalues of S are non-negative). This further implies that $\Lambda(S)/\lambda_{max}(S) \in [0, 1]$, where $\Lambda(S)$ represents any eigenvalue of S.

The idea underlying diffusion wavelets is based on decomposing the spectrum of an input matrix into various spectral bands, spanned by basis functions (Coifman and Maggioni, 2006). Let $T = S/\lambda_{max}(S)$. In Figure 2, we construct spectral bands of eigenvalues, whose associated eigenvectors span the corresponding subspaces. Define dyadic spatial scales t_j as

$$t_j = \sum_{t=0}^{j} 2^t = 2^{j+1} - 1, \;\; j \geq 0 \, .$$

At each spatial scale, the spectral band is defined as:

$$\Omega_j(T) = \{\lambda \in \Lambda(T), \lambda^{t_j} \geq \varepsilon\},$$

where $\Lambda(T)$ represents any eigenvalue of T, and $\varepsilon \in (0, 1)$ is a pre-defined threshold in Figure 2. We can now associate with each of the spectral bands a vector subspace spanned by the corresponding eigenvectors:

$$V_j = \langle \{\xi_\lambda : \lambda \in \Lambda(T), \lambda^{t_j} \geq \varepsilon\} \rangle, \;\; j \geq 0 \, .$$

In the limit, we obtain

$$\lim_{j \to \infty} V_j = \langle \{\xi_\lambda : \lambda = 1\} \rangle$$

That is, the highest level of the resulting subspace hierarchy is spanned by the eigenvector associated with the largest eigenvalue of T. □

This result shows that the multiscale analysis of the relation space will automatically terminate at the level spanned by one basis, which is the most popular relation topic in the training data.

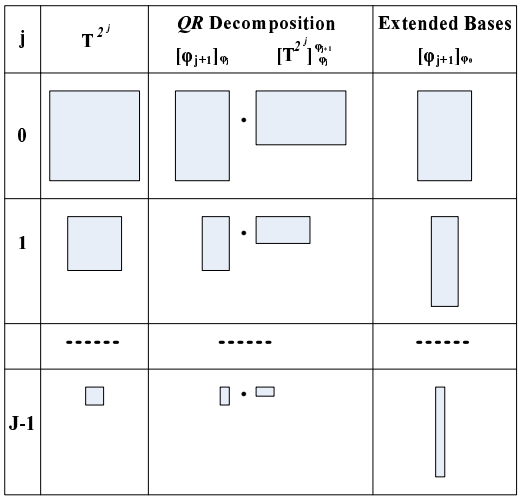

Figure 3: Learning Relation Topics at Multiple Scales.

3.3 High Level Explanation

We first create a set of rules to characterize each input relation. Since these rules may occur in multiple relations, they provide a way to model the co-occurrence relationship between different relations. Our algorithm starts with the relation co-occurrence matrix and then repeatedly applies QR decomposition to learn the topics at the current level while at the same time modifying the matrix to focus more on low-frequency indirect co-occurrences (between relations) for the next level. Running $\mathcal{DWT}$ is equivalent to running a Markov chain on the input data forward in time, integrating the local geometry and therefore revealing the relevant geometric structures of the whole data set at different scales. At scale j, the representation of $T^{2^{j+1}}$ is compressed based on the amount of remaining information and the desired precision. This procedure is illustrated in Figure 3. In the resulting topic space, instances with related relations will be grouped together. This approach may significantly help us detect new relations, since it potentially expands the information brought in by new relation instances from making use of the knowledge extracted from the existing relation repository.

3.4 Benefits

As shown in Figure 3, the topic spaces at different levels are spanned by a different number of basis functions. These numbers reveal the dimensions of the relevant geometric structures of data at different levels. These numbers are completely data-driven: the diffusion wavelets approach can automatically find the number of levels and simultaneously generate the topics at each level. Experiments show that most multiscale topics are interpretable (due to the sparsity of the scaling functions), such that we can interpret the topics at different scales and select the best scale for embedding. Compared to bootstrapping approach, our approach is accumulative; that is as the system learns more relations, it gets better at learning new relations. Because our approach takes advantage of the previously learned relations, and the topic space is enriched as we learn more and more relations.

We use diffusion wavelets (DWT) rather than other hierarchy topic models like hLDA (Blei et al., 2004) to extract relation topics for two reasons. First, DWT is parameter free while other models need some user-input parameters like hierarchy level. Second, DWT is more efficient than the other models. After the relation correlation matrix is constructed, DWT only needs a couple of minutes to extract multiscale topics on a regular computer. A direct experimental comparison between DWT and hLDA can be found in (Wang and Mahadevan, 2009).

4 Constructing Relation Detectors with Multiscale Relation Topics

4.1 Project Relation Instances onto Topics

When we design detectors for new relations, we treat arg_1, arg_2, $noun$, and $verb$ separately to get stronger correlations between relations. We do not directly use $preposition$. Any DBpedia relation $r \in \{1, \cdots, M\}$ is represented with 4 vectors $r_t = [r_t(1), \cdots, r_t(N_t)]$, where $t \in \{arg_1, arg_2, noun, verb\}$, N_t represents the size of the vocabulary set of the type t component in the Wikipedia training data, and $r_t(j)$ represents the occurrence count of type t component in relation r. For example, N_{verb} is the size of the verb vocabulary set in the training data and $r_{verb}(j)$ represents the occurrence count of the j^{th} verb in relation r. When a new relation instance x is given, we extract the dependency path between two arguments, and create four vectors x_t, where $t \in \{arg_1, arg_2, noun, verb\}$, following the same format as r_t. The projection result of x_t onto the DBpedia relation space X_t is as follows:

$X_t = [< r_t(1), x_t(1) >, \cdots, < r_t(M), x_t(M) >]$, where $< \cdot, \cdot >$ is the cosine similarity of two vectors. At level k, the embedding of x is $E^k_x = [E^k_{X_{arg_1}}, E^k_{X_{arg_2}}, E^k_{X_{noun}}, E^k_{X_{verb}}]$, where $E^k_{X_t} = ([\phi_k]_{\phi_0})^T X_t$, and $[\phi_k]_{\phi_0}$ is defined in Figure 2.

4.2 Design New Kernel Using Topic Features

We combine E^k_x with 3 existing kernels ($K_{Argument}$, K_{Path} and K_{BOW}) to create a new kernel for relation detection.

(1) $K_{Argument}$ matches two arguments, it returns the number of common argument types that the input arguments share.

(2) K_{Path} matches two dependency paths. This kernel is formally defined in (Zhao and Grishman, 2005). We extended this kernel by also matching the common nouns, prepositions and verbs in the dependency paths. We assign weight 1 to verbs, 0.5 to nouns and prepositions.

(3) K_{BOW} models the number of common nouns, prepositions and verbs in the given sentences but not in the dependency paths. Since these words are not as important as the words inside the dependency path, we assign weight 0.25 to them.

(4) $K_{TF_k}(x, y) =< E^k_x, E^k_y >$, where x, y are two input relation instances, and $< \cdot, \cdot >$ models the cosine similarity of two vectors. TF stands for topic feature.

(5) The final kernel used in this paper is

$\alpha_1 K_{Argument} + \alpha_2 K_{Path} + \alpha_3 K_{BOW} + \alpha_4 K_{TF_k}$, where α_i can be tuned for each individual domain. In this paper, we set $\alpha_i = 1$ for $i \in \{1, 2, 3, 4\}$.

4.3 Algorithm to Construct Relation Detectors

1. **Construct a relation repository from Wikipedia.**
   (a) Collect training data from Wikipedia and DBpedia (Section 2.1);
   (b) Clean the data representing each input relation (Section 2.2 and 2.3);
   (c) Create relation correlation matrix S following the approach described in Section 3.2, resulting in an $M \times M$ matrix.
2. **Create multiscale relation topics.**
   $[\phi_k]_{\phi_0} = \mathcal{DWT}(S/\lambda_{max}(S), \varepsilon)$, where $\mathcal{DWT}()$ is the diffusion wavelets implementation described in Section 3.1. $[\phi_k]_{\phi_0}$ are the scaling function bases at level k represented as an $M \times p_k$ matrix, $k = 1, \cdots, h$ represents the level in the topic hierarchy. The value of p_k is determined in $\mathcal{DWT}()$ based on the intrinsic structure of the given dataset. Columns of $[\phi_k]_{\phi_0}$ are used as relation topics at level k.
3. **Construct relation detectors for new relations.**
   Given the training data from a new relation, project the data onto level k of the multiscale topic hierarchy, where k is chosen by users (Section 4.1). Apply SVM classifiers together with our kernel (Section 4.2) to create detectors for new relations.

5 Experimental Results

We used SVMLight (Joachims, 1999) together with the user defined kernel setting in our approach. The trade-off parameter between training error and margin c is 1 for all experiments. Our approach to learn multiscale relation topics is largely parameter free. The only parameter to be set is the precision $\varepsilon = 10^{-5}$, which is also the default value in the diffusion wavelets implementation.

5.1 Learning Multiscale Relation Topics

Following the approach discussed in Section 2.1, we collect more than 620,000 training instances for

Table 1: Number of topics at different levels (DBpedia Relations) under 5 different settings: use args, noun, preposition and verb; arg_1 only; arg_2 only; noun only and verb only.

Level	args & words	arg_1	arg_2	noun	verb
1	7628	7628	7628	7628	7628
2	269	119	155	249	210
3	32	17	19	25	35
4	7	5	5	7	10
5	3	2	3	4	4
6	2	1	2	2	2
7	1		1	1	1

7,628 DBpedia relations. For any given topic vector v, we know it is a column vector of length M, where M is the size of the DBpedia relation set and $\|v\|$ = 1. The entry $v[i]$ represents the contribution of relation i to this topic. To explain the main concept of topic v, we sort the entries on v and print out the relations corresponding to the top entries. These relations summarize the topics in the relation repository. One topic example is as follows: [*doctoraladvisor (0.683366), doctoralstudents (0.113201), candidate (0.014662), academicadvisors (0.008623), notablestudents (0.003829), college (0.003021), operatingsystem (0.002964), combatant (0.002826), influences (0.002285), training (0.002148),* $\cdots$], where $doctoraladvisor$ is a DBpedia relation and 0.683366 is its contribution to the topic. The length of this relation vector is 7,628. We only list the top 10 relations here.

Our approach identifies 5 different topic hierarchies under different settings (use args, noun, preposition and verb; arg_1 only; arg_2 only; noun only and verb only). The number of the topics at each level is shown in Table 1. At the first level, each input relation is treated as a topic. At the second level, numbers of topics go down to reasonable numbers like 269. Finally at the top level, the number of topic is down to 1 (Theorem 2 also proves this). We show some topic examples under the first setting. The 3 topics at level 5 are shown in Table 2. They represent the most popular DBpedia relation topics. Almost all 269 topics at level 5 look semantically meaningful. They nicely capture the related relations. Some examples are in Table 3.

Table 2: 3 topics at level 5 (all word types and args).

Top 4 Relations and Their Contributions
starring 86.6%, writer 3.8%, producer 3.2%, director 1.6%
birthplace 75.3%, clubs 6.1%, deathplace 5.1%, location 4.1%
clubs 55.3%, teams 9.3%, nationalteam 6.3% college 6.0%

Table 3: Some topics at level 2 (all word types and args).

Top Relations
activeyearsenddate, careerend, finalyear, retired
commands, partof, battles, notablecommanders
occupation, shortdescription, profession, dates
influenced, schooltradition, notableideas, maininterests
destinations, end, through, posttown
prizes, award, academyawards, highlights
inflow, outflow, length, maxdepth
after, successor, endingterminus
college, almamater, education

5.2 Relation Detection on Wikipedia Data

In previous experiment, 20,000 relation instances were held and not used to construct the topic space. These instances are randomly selected from 100 relations (200 instances from each relation). This set is used as a benchmark to compare different relation detection approaches. In this experiment, 100 instances from each relation are used for training, and the other 100 are for testing. In training, we try three different settings: $n = 5$, 20 and 100, where n is the size of the training set for each relation. When we train a model for one relation, we use the training positive instances from the other 99 relations as training negatives. For example, we use 5 training positive instances and 5*99=495 training negatives to train a detector for each relation.

We compare our approach against the regular rule-based approach (Lin and Pantel, 2001) and two other kernel-based approaches (presented in Section 4.2) for relation detection task. The comparison results are summarized in Table 4. The approach using relation topics (level 2) consistently outperforms the other three approaches in all three settings. When $n = 5$, it achieves the largest improvement over the other three. This indicates that using relation topics that integrate the knowledge extracted from the existing relations, can significantly benefit us when the training data is insufficient. This is reasonable, since the prior knowledge becomes more valuable in this scenario.

The users can select the level that is the most appropriate for their applications. In this example, we only have alignment results at 7 levels. Choosing the space at level 2 spanned by a couple of hundreds of basis functions is a natural choice, since the levels below and above this have too many or too few features, respectively. A user can also select the most appropriate level by checking if the related relation topics are meaningful for their applications.

5.3 Relation Detection on ACE Data

In this experiment, we use the news domain documents of the ACE 2004 corpus (ACE, 2004) to compare our approaches against the state-of-the-art approaches. This dataset includes 348 documents and around 4400 relation instances. 7 relation types, 7 entity types, numerous relation sub-types, entity sub-types, and mention types are defined on this set. The task is to classify the relation instances into one of the 7 relation types or "NONE", which means there is no relation. For comparison, we use the same setting as (Zhang et al., 2006), by applying a 5-fold cross-validation. The scores reported here are the average of all 5 folds. This is also how the other approaches are evaluated. In this test, we treat entity types, entity sub-types and mention types equally as argument types. Table 5 summarizes the performance after applying the kernels presented in Section 4.2 incrementally, showing the improvement from each individual kernel. We also compare our approaches to the other state-of-the-art approaches including Convolution Tree kernel (Collins and Duffy, 2001), Syntactic kernel (Zhao and Grishman, 2005), Composite kernel (linear) (Zhang et al., 2006) and the best kernel in (Nguyen et al., 2009). Our approach with relation topics at level 2 has the best performance, achieving a 73.24% F-measure. The impact of the relation topics is huge. They improve the F-measure from 61.15% to 73.24%. We also test our approach using the topics at level 3. The performance is slightly worse than using level 2, but still better than the others.

This paper studies how relation topics extracted from Wikipedia relation repository can help improve relation detection performance. We do not want to tune our approach to one particular relation detection task, like ACE 2004. In our experiments, no parameter tuning was taken and no domain specific heuristic rules were applied. We are aware of some methods that could stack on our approach to further improve the performance on ACE test. The Composite kernel result in Table 5 is based on a linear combination of the Argument kernel and Convolution Tree kernel. (Zhang et al., 2006) showed that by carefully choosing the weight of each component and using a polynomial expansion, they could achieve the best performance on this data: 72.1% F-measure. (Nguyen et al., 2009) further showed that the performance can be improved by taking syntactic and semantic structures into consideration. They used several types of syntactic information including constituent and dependency syntactic parse trees to improve the state of the art approaches to 71.5% on F-measure. Heuristic rules extracted from the target data can also help improve the performance. (Jiang and Zhai, 2007) reported that by taking several heuristic rules they can improve the F-measure of Composite Kernel to 70.4%. They also showed that using maximum entropy classifier rather than SVM achieved the best performance on this task: 72.9% F-measure. To the best of our knowledge, the most recent result was reported by (Zhou and Zhu, 2011), who extended their previous work in (Zhou et al., 2007). By using several heuristics to define an effective portion of constituent trees, and training the classifiers using ACE relation sub-types (rather than on types), they achieved an impressive 75.8% F-measure. However, as pointed out in (Nguyen et al., 2009), such heuristics are tuned on the target relation extraction task and might not be appropriate to compare against the automatic learning approaches. Even though we have not done any domain specific parameter tuning or applied any heuristics, our approach still achieve significant improvements over all approaches mentioned above except one, which is based on heuristics extracted from the target domain. This also implies that by combining some of the above ideas with relation topics, the performance on ACE data may be further improved.

6 Conclusions

This paper proposes a novel approach to create detectors for new relations integrating the knowledge extracted from the existing relations. The contributions of this paper are three-fold. Firstly, we pro-

Table 4: F-measure comparison of different approaches over 100 DBpedia relations with 5, 20 and 100 positive examples per relation. AG: $K_{Argument}$, DP: K_{Path}, BOW: K_{BOW}, TF_k: K_{TF_k}.

Approaches	100	20	5
Rule Based	37.70%	27.45%	13.20%
AG+ DP	73.64%	51.85%	22.95%
AG+ DP+ BOW	78.74%	62.76%	31.98%
AG+ DP+ BOW+ TF_2	81.18%	68.03%	41.60%

Table 5: Performance comparison of different approaches with SVM over the ACE 2004 data. P: Precision, R: Recall, F: F-measure, AG: $K_{Argument}$, DP: K_{Path}, BOW: K_{BOW}, TF_k: K_{TF_k}.

Approaches	P(%)	R(%)	F(%)
Convolution Tree Kernel	72.5	56.7	63.6
Composite Kernel (linear)	73.50	67.00	70.10
Syntactic Kernel	69.23	70.50	69.86
Nguyen, et al. (2009)	76.60	67.00	71.50
AG	59.56	46.22	52.02
AG + DP	64.44	54.93	59.28
AG + DP + BOW	62.00	61.19	61.15
AG + DP + BOW + TF_3	69.63	76.51	72.90
AG + DP + BOW + TF_2	69.15	77.88	73.24

vide an automatic way to collect training data for more than 7,000 relations from Wikipedia and DBpedia. Secondly, we present an unsupervised way to construct a set of relation topics at multiple scales. Different from the topics defined over words, relation topics are defined over the existing relations. Thirdly, we design a new kernel for relation detection by integrating the relation topics in the representation of the relation instances. By leveraging the knowledge extracted from the Wikipedia relation repository, our approach significantly improves the performance over the state-of-the-art approaches on ACE data. This paper makes use of all DBpedia relations to create relation topics. It is possible that using a subset of them (more related to the target relations) might improve the performance. We will explore this in future work.

Acknowledgments

We thank the reviewers for their helpful comments. This material is based upon work supported in part by the IBM DeepQA (Watson) project. We also gratefully acknowledge the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0172. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the DARPA, AFRL, or the US government.

References

ACE. 2004. The automatic content extraction projects, http://projects.ldc.upenn.edu/ace/.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference, Busan, Korea*, pages 11–15. Springer.

D. Blei, A. Ng, and M. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. 2004. Hierarchical topic models and the nested Chinese restaurant process. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 152–160.

R. Coifman and M. Maggioni. 2006. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21:53–94.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 625–632.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 423–429.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland,

Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165:91–134.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. 2010. Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 286–295.

Jing Jiang and Chengxiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 113–120.

Jing Jiang. 2009. Multi-task transfer learning for weakly-supervised relation extraction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL) and the 4th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1012–1020.

T. Joachims. 1999. *Making Large-Scale SVM Learning Practical*. MIT Press.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*.

Dekang Lin and Patrick Pantel. 2001. DIRT - discovery of inference rules from text. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 323–328.

Michael McCord. 1995. Slot grammar: A system for simpler construction of practical natural language grammars. *Communications of the ACM*, 38(11).

Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL) and the 4th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1003–1011.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

B. Schölkopf and A. J. Smola. 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. YAGO: A large ontology from Wikipedia and WordNet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217.

C. Wang and S. Mahadevan. 2009. Multiscale analysis of document corpora based on diffusion models. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1592–1597.

Wikipedia. 2011. http://www.wikipedia.org/.

Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 419–426.

G. Zhou and Q. Zhu. 2011. Kernel-based semantic relation detection and classification via enriched parse tree structure. *Journal of Computer Science and Technology*, 26:45–56.

G. Zhou, M. Zhang, D. Ji, and Q. Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*.