Knowledge and Information Systems (2019) 61:1695–1713
https://doi.org/10.1007/s10115-018-1250-6

REGULAR PAPER

Categorizing relational facts from the web with fuzzy rough sets

Aditya Bharadwaj[1] · **Sheela Ramanna**[1]

Received: 16 October 2017 / Revised: 13 May 2018 / Accepted: 22 July 2018 / Published online: 31 August 2018

Abstract
Significant advances have been made in automatically constructing knowledge bases of relational facts derived from web corpora. These relational facts are linguistic in nature and are represented as ordered pairs of nouns (Winnipeg, Canada) belonging to a category (City_Country). One major problem is that these facts are abundant but mostly unlabeled. Hence, semi-supervised learning approaches have been successful in building knowledge bases where a small number of labeled examples are used as seed (training) instances and a large number of unlabeled instances are learnt in an iterative fashion. In this paper, we propose a novel fuzzy rough set-based semi-supervised learning algorithm (FRL) for categorizing relational facts derived from a given corpus. The proposed FRL algorithm is compared with a tolerance rough set-based learner (TPL) and the coupled pattern learner (CPL). The same ontology derived from a subset of corpus from never ending language learner system was used in all of the experiments. This paper has demonstrated that the proposed FRL outperforms both TPL and CPL in terms of precision. The paper also addresses the concept drift problem by using mutual exclusion constraints. The contributions of this paper are: (i) introduction of a formal fuzzy rough model for relations, (ii) a semi-supervised learning algorithm, (iii) experimental comparison with other machine learning algorithms: TPL and CPL, and (iv) a novel application of fuzzy rough sets.

Keywords Text categorization · Relational facts · Semi-supervised learning · Fuzzy rough sets · Web mining

This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant. Special thanks to Cenker Sengoz for sharing the data set and for discussions regarding TPL. We are very grateful to Prof. Estevam R. Hruschka Jr. for the NELL dataset.

✉ Sheela Ramanna
s.ramanna@uwinnipeg.ca

Aditya Bharadwaj
bharadwaj-a@webmail.uwinnipeg.ca

[1] Department of Applied Computer Science, University of Winnipeg, Winnipeg, Canada

1 Introduction

Information extraction is the task of automatically extracting knowledge from text and constructing knowledge bases. A knowledge base usually contains entities such as Tom Brady, Winnipeg, or Manchester United FC. It also contains information about these entities, such as the *fact* that Tom Brady plays football, Winnipeg is a city in Canada, or Manchester United FC is a sports team. These facts are essentially linguistic in nature and referred to commonly as nouns (or categorical instances) or relations (which are ordered pairs of nouns). Examples of *nouns* are Winnipeg, Warsaw, New Delhi which belong to the *category* of City. Examples of *relations* which are ordered pairs of nouns are: (Winnipeg, Canada), (Warsaw, Poland), (New Delhi, India), belonging to the *category* of CityInCountry. Knowledge bases are utilized in many applications such as machine translation, question answering, and semantic search [28]. The growth of the web has spawned knowledge bases from web corpora where the construction of these bases is performed in a either semi-supervised or unsupervised manner. These methods require minimal or no human intervention and can recursively discover new relations, attributes, and instances in a fully automated, scalable manner. The focus of some of the current systems such as YAGO3 [17,35,36], KnowItAll [11] and TextRunner [1] is on *machine reading*. Other systems such as NELL [4,19] successfully learn facts over a prolonged period of time and are called *never ending learners*.

Typical *web corpora* include Wikipedia, DBpedia, WordNet [18], NELL [19], and Google's Knowledge Vault [9]. In this work, inspired by NELL, we focus on a small subset of the problem where facts are learned using machine learning techniques in an iterative fashion rather than machine reading or the extraction process itself. In other words, we are given facts that have already been extracted and new facts are learnt. Content for such learning can be thought of as a triple (subject, predicate, object). For example, a *relational fact* can be represented as:

$$\langle \text{Tom Brady} \rangle \ \langle \text{actually led} \rangle \ \langle \text{Patriots} \rangle \ \text{to a win}$$

where "Tom Brady" is a subject (noun), "actually led" is a predicate, and "Patriots" is an object (noun). This representation permits the learning of a relational fact (Tom Brady, Patriots) as belonging to a category (Athlete-team) using the predicate information. Unary relations (nouns) capture memberships in a semantic type. On the other hand, binary relations (a pair of nouns) capture semantics between entities. The predicate is referred to as a contextual pattern which is an arbitrary phrase such as "actually led" providing a *context* for the relation. Hence, we have two main facts: relations and contextual patterns. A matrix is then formed based on the co-occurrence of relations and their contextual patterns. Using this co-occurrence matrix, if a relation r_1 (e.g., Tom Brady, Patriots) regularly co-occurs in context (e.g., "actually led") along with a relation r_2 (e.g., Peyton Manning, Colts) and if r_1 belongs to category C (e.g., Athlete_Team), then it is *likely* that r_2 might also belong to category Athlete_Team. In this paper, we address three typical issues that arise from this form of learning: i) the number of training examples are few, i.e., relations and their known categories, ii) a relation may belong to more than one category depending on its contextual patterns, and iii) new relations end up being miscategorized (also known as concept drift [7,12]) in the process of never ending learning.

To address the *first* issue, semi-supervised approaches are preferred where a small number of labeled examples are used as seed (training) examples and a large number of unlabeled instances are learnt in an iterative fashion. In every iteration, a few examples from the unlabeled instances are "promoted" as trusted, thus growing the knowledge base. This process is also known as bootstrapping. Significant improvements in accuracy for learning categories

of nouns and relations have been demonstrated using bootstrapping methods highlighted in [3,4,19,39]. The *second* issue translates to the challenge of using the co-occurrence information and scoring the degree of *belonging* where a relation can belong to more than one category based on its context. In CPL [4], simultaneous training of many extractors for learning relations using mutual exclusion and type-checking relationships were employed. In CBS [39], a probabilistic score for every candidate noun fact belonging to a category was used. In TPL [31], a scoring mechanism based on tolerance approximation spaces [33] derived from rough set theory [24] was used. The foundation for TPL is a tolerance relation which is symmetrical and reflexive, but need not be transitive, and this property makes it possible to represent overlapping classes and well-suited for constructing knowledge bases of linguistic entities. In the proposed research with FRL, instead of a crisp co-occurrence matrix used by the above three methods, our method uses a fuzzy (graded) co-occurrence matrix. The motivation for using a hybrid rough set method was to take advantage of the strengths of both fuzzy and rough set methods. Fuzzy rough model for categorizating *nouns* which permits the modeling of the degree of belonging in a more powerful manner was successfully used in [2]. In this paper, a scoring mechanism is used based on fuzzy rough sets [10], where fuzzy sets [42] are first applied to the co-occurrence information. Rough set operators are then applied to the *graded* co-occurrence information to obtain the overall score. Such a technique was successfully employed in [8] for web query expansion for document retrieval.

The *third* issue of the concept drift problem is common to all iterative semi-supervised learning, where the limited number of initial labeled examples tends to be insufficient to reliably constrain the learning process, thereby causing the prediction of new relational facts to become less accurate with each iteration. CPL relies on three types of constraints to handle concept drift. In CBS, Bayesian learning is used to learn several categories for nouns simultaneously using a coupled Bayesian sets algorithm with mutual exclusion constraints. With the tolerance rough set-based learner (TPL) algorithm [29], no external constraints were required for learning nouns and relations. In the proposed FRL method, mutual exclusion constraints are defined to handle concept drift.

In this paper, we present a new hybrid fuzzy rough model inspired by [6,8] to learn relations. We introduce a semi-supervised learning algorithm (FRL) with ontology derived from a subset of examples from NELL. The proposed FRL algorithm is experimentally compared with CPL and TPL algorithms. Experimental results demonstrate that FRL performs better than TPL based on the ranking method and outperforms CPL and TPL using the promotion-based method. The contributions of this paper are: (i) a formal fuzzy rough model for relations, (ii) a semi-supervised learning algorithm, (iii) experimental comparison with other machine learning algorithms: TPL and CPL, and (iv) a novel application of fuzzy rough sets with some insights into the strengths and weakness of integration of these technologies for categorization of linguistic entities.

The paper is organized as follows: in Sect. 2, we discuss research related to structured (document) and unstructured text (linguistic) categorization pertinent to this paper. In this section, we describe TPL, CBS, and CPL in some detail. In addition, we also discuss the fuzzy rough sets model for documents. In Sect. 3, we present the proposed fuzzy rough sets framework for nouns and relations. In Sect. 4, we describe our proposed FRL algorithm and experiments followed by a trace of FRL with examples. In Sect. 5, we discuss the experimental setup and analysis of the results. Here, we illustrate the problem of concept drift with FRL. In Sect. 6, we discuss scalability and complexity of FRL. We conclude the paper in Sect. 7.

2 Related works

In this section, we briefly introduce CPL and CBS methods since they are used as benchmarks for comparison. Tolerance rough set method is discussed in considerable detail, and the TPL framework is introduced to give the reader an insight into the lower and upper approximation framework. We also discuss the fuzzy rough set method which was first proposed as a model for document representation and retrieval and uses a graded (fuzzy) thesaurus.

2.1 CPL and CBS methods

CPL is a bootstrapping algorithm based on logistic regression that uses mutual exclusion and type-checking constraints [4]. Nouns and relations are first filtered to enforce mutual exclusion and type-checking constraints. Next, for each category, CPL ranks the linguistic entities using the number of promoted patterns that they co-occur with, so that candidates that occur with more patterns are ranked higher.

CBS is based on Bayesian sets [13] and uses the co-occurrence statistics between noun phrases and contextual patterns. CBS calculates a probabilistic score by using those co-occurrence statistics for every candidate category. In CBS, the learned functions can be considered as classifiers that enforce mutual exclusion constraints using positive examples of one category as negative examples for other ones to learn high-precision instances for all categories defined in an initial ontology. In CBS, instances are first filtered to enforce mutual exclusion. Then, the top ranked ones are promoted as trusted instances for that category. The promoted instances are used as seeds in subsequent iterations in a semi-supervised iterative manner. CBS learns several categories simultaneously. Readers can refer to [39] for more details.

2.2 Tolerance rough sets method

Rough set theory consists of an approximation space characterized by an equivalence relation [24]. Learning in rough set theory is typically accomplished using two operators: lower and upper approximation. However, it has been demonstrated that a tolerance relation [25,33] is more appropriate for document and named entity classification rather than an equivalence relation [16,27,40]. This is because a tolerance relation is symmetrical and reflexive, but need not be transitive, and this property makes it possible to represent overlapping classes which is ideal for text categorization. Tolerance rough sets became the basis for document representation and clustering starting from 2000 [14,16,20–22,38]. In [32], a tolerance-based semi-supervised two-class ensemble classifier for documents was proposed. In [37], a weighting scheme for the tolerance rough sets model based on neural networks was introduced. A lexicon-based document representation (LBDR) with tolerance rough sets was introduced by Virginia et al. [40] and subsequently elaborated in [41]. In [27,29–31], tolerance rough sets were used for the first time in categorizing nouns and relational facts.

A tolerance approximation space $\mathcal{A}$ [33] is denoted by $\mathcal{A} = (U, I, \nu)$ where U is the universe of objects, I defines the *tolerance class* of an object in the universe. In other words, it defines the neighborhood of each object, and ν measures the *degree of inclusion* between two sets. These objects can be documents, nouns, relations or contextual patterns. We now briefly show how these parameters are used with the lower ($\mathcal{L}_{\mathcal{A}}$) and upper approximators ($\mathcal{U}_{\mathcal{A}}$) in the similarity scoring mechanism for nouns by the tolerant pattern learner (TPL) algorithm.

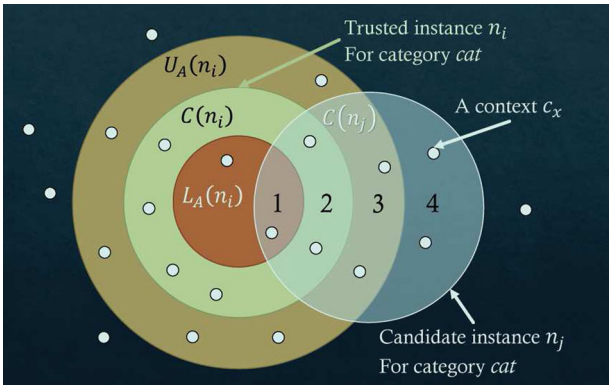

Fig. 1 Four zones of recognition for contexts emerging from approximations of n_i [29]

2.2.1 Tolerant pattern learner (TPL) framework

TPL learns by separating the approximation space for *nouns* into four zones numbered one through four shown in Fig. 1. Each zone contributes toward a different degree of similarity for an instance n_i to be promoted as *trusted*. These instances are shown as *white dots*. Every trusted noun instance n_i of a given category cat is associated with three descriptor sets: $C(n_i)$, $\mathcal{U}_\mathcal{A}(n_i)$ and $\mathcal{L}_\mathcal{A}(n_i)$ where $C(n_i)$ is a function that is used to determine all the *co-occurring* contexts for noun n_i. The fourth zone (not named) represents one that does not contribute to the similarity determination and may be considered as a negative zone.

The lower and upper approximators are calculated as follows using the inclusion function ν and uncertainty function I which defines the neighborhood using a threshold of θ:

$$\mathcal{L}_\mathcal{A}(n_i) = \{c_j \in \mathcal{C} : \nu(I_\theta(c_j), C(n_i)) = 1\}$$
$$\mathcal{U}_\mathcal{A}(n_i) = \{c_j \in \mathcal{C} : \nu(I_\theta(c_j), C(n_i)) > 0\}$$

A *micro-score* for the candidate noun phrase n_j against the trusted instance n_i of the category cat is calculated as follows:

$$\text{micro}(n_i, n_j) = \omega(C(n_i), C(n_j))\alpha + \omega(\mathcal{U}_\mathcal{A}(n_i), C(n_j))\beta + \omega(\mathcal{L}_\mathcal{A}(n_i), C(n_j))\gamma$$

where ω is an overlap function and the parameters α, β, γ are weights that reflect the degrees of similarity contribution of objects belonging to specific zones.

2.3 Fuzzy rough sets method

Combining rough and fuzzy sets as an efficient soft computing strategy for machine learning was explored in [23]. In fuzzy set theory [42], the relational counterpart generates soft similarity classes which permits partial overlap, even when the fuzzy relation is reflexive, symmetrical and $\mathcal{T}$-transitive, i.e., a so-called fuzzy $\mathcal{T}$-equivalence relation [8]. This is the property that lies at the heart of fuzzy rough set models and lends itself to text categorization applications [5,10]. Fuzzy rough set method was first proposed as a model for structured text representation (document) and retrieval [34]. A fuzzy rough set is a pair $(A_1, A_2) \in (X, R)$ where A is a fuzzy set in X such that $R \downarrow A = A_1$ and $R \uparrow A = A_2$. R is a fuzzy relation in

X [6] where $\downarrow$ represents the lower approximator and $\uparrow$ represents the upper approximator. We now briefly describe the fuzzy rough set framework for documents.

In [8], a thesaurus which consists of web pages (as documents) is generated using a query of two terms t_1 and t_2. Let D_{t_1} and D_{t_2} be the number of pages that contain t_1 and t_2 terms, respectively. Then, the following measure is used to construct an initial thesaurus:

$$\frac{|D_{t_1} \cap D_{t_2}|}{\min |D_{t_1}|, |D_{t_2}|} \tag{1}$$

This initial thesaurus is then normalized by using an S-function which generates the fuzzified thesaurus. Working with fuzzy sets, it is often assumed that the relation characterizing the approximation space is transitive. Therefore, the t-norm $\mathcal{T}$ is used to construct a transitive fuzzy thesaurus where the documents with membership value of 0.5 or above are considered.

A document retrieval is then viewed as a query expansion problem where the operators of rough sets (upper and lower) are combined with a graded (fuzzy) thesaurus. The thesaurus is a fuzzy relation, and the query is a fuzzy set. Specifically, a query is expanded by adding all the related terms (upper approximation) and then pruned using the lower approximation in what is termed as a tight upper approximation. The intuition behind this operation was to counter the disadvantage of using transitive fuzzy thesaurus where certain terms are added to the original query with a high degree of similarity that have no relevance to the original query.

3 Proposed fuzzy rough sets framework for relations

One can view a knowledge base as a thesaurus consisting of linguistic entities such as categorical nouns, relations, and their contextual patterns. In this section, we introduce the model that is used by FRL. For the sake of completeness, we have included definitions for nouns introduced in [2].

3.1 Formal model

- $\mathcal{N} = \{n_1, n_2, \ldots, n_M\}$ is the universe of nouns.
- $\mathcal{C} = \{c_1, c_2, \ldots, c_P\}$ is the universe of categories.
- $\mathcal{R} = \{r_1, r_2, \ldots, r_Q\}$ is the universe of contextual patterns.
- $\mathcal{H} = \{h_{ij} = (n_i, n_j) \in \mathcal{N}^2 : \exists r_k \in \mathcal{R} | f_{\mathcal{H}}(h_{ij}, r_k) > 0\}$ is the universe of relations described via the relational co-occurrence function $f_{\mathcal{H}}(h_{ij}, r_k)$ where $f_{\mathcal{H}}(h_{ij}, r_k) = \{k \in \mathbb{N} : h_{ij}$ occurs k times within the context $r_k\}$.
- $\mathcal{TN} = \{n_1, n_2, \ldots, n_Y\}$ is a set of trusted nouns such that $\mathcal{TN} \subset \mathcal{N}$ and index $Y < M$.
- $\mathcal{TR} = \{h_{ab}, h_{bc}, \ldots, h_{cd}\}$ is a set of *trusted relations* such that $\mathcal{TR} \subset \mathcal{H}$ and index $cd < ij$.

We define the following cross mapping functions which form the basis for the co-occurrence matrix.

- $C : \mathcal{N} \rightarrow \mathbb{P}(\mathcal{C})$ denotes a mapping of each noun to its set of co-occurring noun contexts such that $C(n_i) = \{c_j : f_C(n_i, c_j) > 0\}$ where $f_C(n_i, c_j) = \kappa \in \mathbb{N}$ denoting that n_i occurs κ times within context c_j.
- $R : \mathcal{H} \rightarrow \mathbb{P}(\mathcal{R})$ denote mapping of each relation to its set of *co-occurring relational* contexts: $R(h_{ij}) = \{r_k : f_R(h_{ij}, r_k) > 0\}$

Note that the function $f_R(h_{ij}, r_k)$ is used in the thesaurus construction process.

3.2 Relation thesaurus construction

Based on the definitions introduced in Sect. 3.1, let $\mathcal{H}$ be a set of relations and $\mathcal{R}$ be a set of co-occurring contextual relational patterns. Let $\mathcal{TR}$ be a set of trusted relations such that $\mathcal{TR} \subset \mathcal{H}$. Similar to the procedure given in [2,8], the first step toward creation of a fuzzy thesaurus is by normalizing the co-occurrence information using ϑ:

$$\vartheta(h_{ij}, r_k) = \frac{f_R(h_{ij}, r_k)}{f_R(h_{ij}, r_k), \forall k : 1 \ldots Q} \tag{2}$$

The next step involves fuzzifying the co-occurrence function with the S-function where $\alpha = 0.001$ and $\beta = 0.02$ are constants determined experimentally.

$$S(\vartheta; \alpha, \beta) = \begin{cases} 1 & \text{if } \vartheta \geq \beta \\ \frac{\vartheta - \alpha}{\beta - \alpha} & \text{if } 0.005 \leq \vartheta < \beta \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

3.3 Lower and upper approximation of the relation thesaurus

The initial thesaurus I for relations is defined as an approximation space $I = (\mathcal{H}, \mathcal{R}, \mathrm{CO}_F)$, where $\mathcal{H}$ denotes the universe of relations, $\mathcal{R}$ denotes the co-occurring contextual patterns. The fuzzy relation CO_F is defined as a fuzzy set in $\mathcal{H} \times \mathcal{R}$. Let $\mathcal{H}_\mathcal{F}$, $\mathcal{TR}_\mathcal{F}$ represent the fuzzy sets of relations and trusted relations, respectively. The *upper* and *lower* approximations of the fuzzy set $\mathcal{H}_\mathcal{F}$ in I is denoted by $\mathcal{H}_\mathcal{F} \uparrow \mathrm{CO}_F$ and $\mathcal{H}_\mathcal{F} \downarrow \mathrm{CO}_F$. The notation for upper and lower approximation operators follows the style from [8,15,26]. The upper approximation is defined as:

$$\mathcal{H}_\mathcal{F} \uparrow \mathrm{CO}_F = \sup_{h_{ij} \in \mathcal{H},\ h_{xy} \in \mathcal{TR}} \left(\mathrm{CO}_F(R(h_{ij}), h_{xy}), \mathcal{H}_\mathcal{F}(h_{xy}) : \mathrm{CO}_\mathcal{F}(h_{ij}) \geq \mathrm{CO}_\mathcal{F}(h_{xy})\right) \tag{4}$$

The upper approximation will make it possible to select candidate relations h_{ij} having a membership co-occurrence value either equal to or more than that of the trusted relations h_{xy}. The lower approximation is defined as:

$$\begin{aligned} \mathcal{H}_\mathcal{F} \downarrow \mathrm{CO}_F = \inf_{h_{ij} \in \mathcal{H},\ h_{xy} \in \mathcal{TR}} & (\mathrm{CO}_F(R(h_{ij}), h_{xy}), \mathcal{H}_\mathcal{F}(h_{xy}) \\ & :((h_{ij}, h_{xy}) | R(h_{xy}) \cap R(h_{ij}) \neq \emptyset)) \end{aligned} \tag{5}$$

Here, the lower approximation will make it possible to select candidate relations h_{ij} as trusted, when there is at least one common context with a trusted noun h_{xy}. The micro-score for a candidate relation h_{ij} from $\mathcal{H}$ is calculated as:

$$\mathrm{micro}(h_{ij}) = \omega_1 \left(\mathcal{H}_\mathcal{F} \uparrow \mathrm{CO}_F\right) + \omega_2 \left(\mathcal{H}_\mathcal{F} \downarrow \mathrm{CO}_F\right) \tag{6}$$

where ω_1 and ω_2 are application dependent and in this experiment are set to 50% and 5%, respectively.

3.4 Tight upper approximation for relations

In [8], the benefit of using tight upper approximation was discussed where a term y, will only be added to a query, if all the terms related to y are related to at least one keyword of the query. A similar problem occurs with learning relations, where candidate relations with

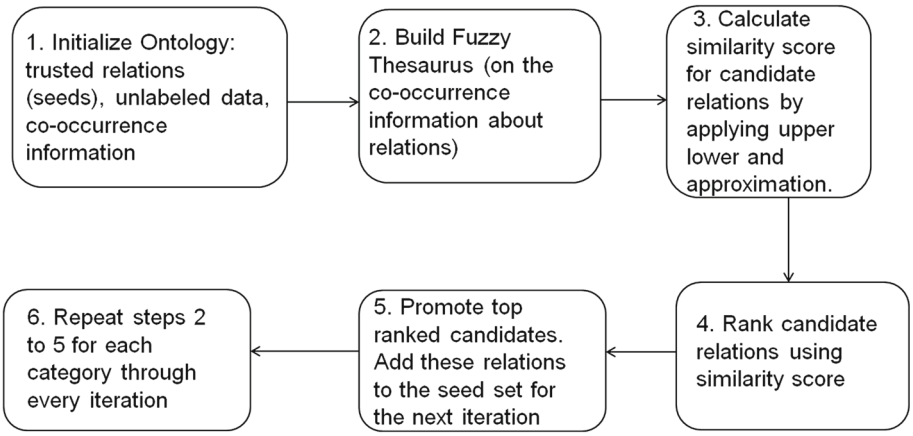

Fig. 2 Semi-supervised learning using FRL

high membership degrees are promoted and added to the set of trusted relations, irrespective of whether the candidate relation is related to the trusted relation. To counter this problem, we take the upper approximation of all candidate relations h_{ij} in $\mathcal{H}$ followed by lower approximation for the remaining candidate relations which results in the following tight upper approximation operation:

$$\mathrm{CO}_F \downarrow\uparrow \mathcal{H}_F(h_{ij}) = \mathrm{CO}_F \downarrow (\mathrm{CO}_F \uparrow \mathcal{H}_F(h_{ij})) \tag{7}$$

We then calculate micro(h_{ij}) given in Eq. 6 for each candidate relation. We sort those relations identified as trusted based on their scores, and the top five relations are promoted as trusted for the next iteration.

4 FRL learning framework

The high-level flow for learning with FRL is shown in Fig. 2. The input to FRL is an ontology of trusted relations (seeds), unlabeled relational facts, and co-occurrence information. Steps 2 and 3 in Fig. 2 implement the fuzzy rough model in determining the similarity score for each candidate relation. The process consists of building a fuzzy thesaurus, applying the upper and lower approximator operators (resulting in tight upper approximation). Steps 4 and 5 form the semi-supervised learning component for promoting instances as trusted and growing the knowledge base. Since learning is done for each category at a time, step 6 shows the iterative nature of this process.

4.1 FRL algorithm

FRL is an iterative algorithm which is designated to run indefinitely.

Examples of input to Algorithm 1 are trusted relations $\mathcal{TR}$ per category such as (e.g., Lionel Messi_FC Barcelona, Michael Jordan_Chicago Bulls), categories (e.g., Athlete_Team) and a large corpus of unlabeled relations $\mathcal{H}$. Another input is a large co-occurrence relational matrix CO representing relations and their co-occurring contextual patterns. The matrix consisted of approximately 6.5 million relations and 11 million contextual patterns.

Algorithm 1: Fuzzy Rough Learner

Input : An ontology O defining categories; a large corpus $\mathcal{H}$, CO co-occurrence matrix, a small set of trusted relations $\mathcal{TR}$
Output: $\mathcal{TR}'$ is a set of all new promoted trusted relations

```
for r = 1 → end of file do
    for each category cat do
        for each new trusted relations h_xy belonging to cat do
            for each candidate relation h_ij do
                Calculate Fuzzy Relation CO_F using Eqs. 2 and 3;
                Calculate Upper Approximation H_F ↑ CO_F as in Eq. 4;
                Calculate score ω1;
                for each candidate relation h_ij do
                    Calculate Lower Approximation H_F ↓ CO_F using Eq. 5;
                    Calculate score ω2;
            Calculate micro_cat(h_ij) using Eq. 6;
        Sort trusted instances h_xy by micro_cat/|cat|;
        Promote top trusted instances, such that TR' = TR ∪ {h_xy};
```

The output of the FRL algorithm is a set of trusted relations categorized by their respective categories from the ontology. We use a score-based ranking. For the category cat, after calculating the score for every candidate of $\mathcal{H}$, we rank the candidates by their micro-scores normalized by the number of trusted instances of cat. Finally, we promote the top new candidates as trusted. After every iteration, FRL learns new trusted instances and grows its knowledge base to make decisions in the subsequent iterations.

4.2 Illustration

In this section, we trace the steps of FRL starting with the examples of categories, trusted relations and contextual patterns which are input to the proposed FRL. Note that a relation is a pair of nouns. The initial seeds are identical to the ones used in TPL and CBS algorithms.

FRL Algorithm-Input
Example: Categories
{Athlete_Team, CEO_Company, City_Country, City_State, Coach_Team, Company_City, Stadium_City, State_Capital, State_Country, Team_vs_Team}

Example: Trusted relations
{FC Barcelona‖Lionel Messi, Chelsea‖Didier Drogba, AC Milan‖Ronaldinho, Chelsea‖Frank Lampard, Bastian Schweinsteiger‖Bayern Munich}

Example: Contextual patterns
{attacking midfielder, forward, have handed, player, prodigy}

Algo. Step 2
In this trace, we consider *category* Athlete_Team. We will be focusing on one *trusted relation* FC Barcelona ‖Lionel Messi, and two *candidate relations* England‖Steven Gerrard and Tiger Woods‖United States.

Algo. Step 3 [Trusted Relations $\mathcal{TR}$]
We start the trace with 5 trusted *relations* which are a pair of nouns separated by‖given as follows:

Table 1 Trusted relations along with their commonly co-occurring contextual patterns

Trusted nouns	Attacking midfielder	Forward	Have handed	Player	Prodigy
FC Barcelona‖Lionel Messi	1	1	2	2	1
Chelsea‖Didier Drogba	1	9	2	2	1
AC Milan‖Ronaldinho	3	2	2	2	1
Chelsea‖Frank Lampard	3	1	2	2	1
Bastian Schweinsteiger‖Bayern Munich	3	1	2	2	1

Table 2 Normalized frequency ϑ for each relation

Trusted nouns	Attacking midfielder	Forward	Have handed	Player	Prodigy
FC Barcelona‖Lionel Messi	1/7	1/7	2/7	2/7	1/7
Chelsea‖Didier Drogba	1/15	9/15	2/15	2/15	1/15
AC Milan‖Ronaldinho	3/10	2/10	2/10	2/10	1/10
Chelsea‖Frank Lampard	3/9	1/9	2/9	2/9	1/9
Bastian Schweinsteiger‖Bayern Munich	3/9	1/9	2/9	2/9	1/9

– FC Barcelona‖Lionel Messi, Chelsea‖Didier Drogba, AC Milan‖Ronaldinho, Chelsea‖Frank Lampard, Bastian Schweinsteiger‖Bayern Munich

We start the trace with the following 5 initial *contextual patterns* for the trusted relations given as follows:

– attacking midfielder, forward, have handed, player, prodigy

Algo. Step 4
We must first fuzzify the co-occurrence information for the trusted relations. The next three steps (4.1 to 4.3) illustrate how the thesaurus is constructed.

Algo. Step 4.1 [Co-occurrence frequency f_R].
Table 1 shows the frequencies of the trusted relations and their co-occurring contextual patterns.

Algo. Step 4.2 [Normalized frequency ϑ].
Based on the co-occurrence information given in Table 1, we normalize the table by calculating the normalized frequency ϑ for each relation. Table 2 shows the normalized result for each trusted relation.

Algo. Step 4-revisited In this step, each iteration starts with only one trusted relation. Hence, we start the trace with FC Barcelona‖Lionel Messi listed in Table 3. This in turns leads to step 4.3.

Algo. Step 4.3 [Calculating membership value S based on Eq. 3].
We use Eq. 3 for all *contextual patterns* for one *trusted relation* FC Barcelona‖Lionel Messi given in Table 3. Note that the S-function values are all one since $\vartheta \geq \beta$ where $\beta = 0.02$ as shown in Table 4.

Algo. Step 5 The above process is repeated for all relations in $\mathcal{TR}$ (trusted relations) which results in $\text{CO}_{\mathcal{F}}$ (fuzzified co-occurrence).
Once we have the membership values for each of the trusted relations, we start the process for each *candidate relation*. A candidate relation is a new relation that must be classified.

Table 3 Normalized frequency ϑ for FC Barcelona||Lionel Messi

FC Barcelona‖Lionel Messi	Normalized frequency ϑ
Attacking midfielder	1/7
Forward	1/7
Have handed	2/7
Player	2/7
Prodigy	1/7

Table 4 *S*-function value for FC Barcelona||Lionel Messi

FC Barcelona‖Lionel Messi	*S*-function
Attacking midfielder	1
Forward	1
Have handed	1
Player	1
Prodigy	1

Table 5 *S*-function value for Steven Gerrard||England

Steven Gerrard‖England	*S*-function
Attacking midfielder	1
Forward	0
Have handed	1
Player	1
Prodigy	1

We repeat steps **Algo. Step 1** to **Algo. Step 5** for each candidate relation and calculate their memberships along with the commonly co-occurring contextual patterns.

Algo. Step 6 Calculate Upper Approximation.

It is the set of all the *candidate relations* having a membership co-occurrence value either equal to or more than that of the *trusted relations*. The threshold ε_u for this dataset was set to 50%. The following members constitute a sample belonging to the upper approximation set. These relations are possible candidates for promotion:
{Steven Gerrard‖England, Tiger Woods‖United States, Ronaldo‖Manchester United, Michael Jordan‖Chicago Bulls, Kobe Bryant‖LA Lakers,…}

Algo. Step 7 Calculate Weight ω_1.

Here, we give the illustration in terms of two candidate relations (Steven Gerrard‖England and Tiger Woods‖United States) from the upper approximation. Note ω_1 is determined based on criteria set in Eq. 4. Table 5 shows the *S*-function values for the candidate relation Steven Gerrard‖England. The contextual patterns listed are determined as common with the trusted patterns.

The weight ω_1 for Steven Gerrard‖England is calculated as the sum of the memberships of the above patterns. Thus, $\omega_1 = 4$. Table 6 gives the *S*-function values for the candidate relation Tiger Woods‖USA.

For the second candidate relation Tiger Woods‖USA, $\omega_1 = 3$.

Algo. Step 9 Calculate lower approximation. Here we apply the lower approximation operator on the set of relations in the upper approximation. First, we find all the contextual patterns for

Table 6 *S*-function value for Tiger Woods‖USA

Tiger Woods‖USA	*S*-function
Attacking midfielder	0
Forward	0
Have handed	1
Player	1
Prodigy	1

all trusted relations. Nineteen contextual patterns for all trusted relations were found by FRL. We only list 14 of them. The remaining 5 contextual patterns are derived from Table 1. A candidate relation is required to have at least one common contextual pattern with the trusted seed relation. The threshold ε_l for this set will be 5% of 19. In other words, a candidate not having at least one common contextual pattern will automatically be eliminated once we apply the lower approximation operator. The following is a list of the 14 contextual patterns:

– "about the signing of", "along with", "are missing", "as striker", " Goal in", "ace a", "added to", "and England footballer", "and England midfielder", " believes", "have confirmed-midfielder", "prodigy" , "winger"

Algo. Step 10: Calculate Weight ω_2 for the two candidate relations shown in two steps. ω_2 is calculated as the sum of the total number of common contextual patterns between a candidate relation and trusted relations. Note, ω_2 is determined based on criteria set in Eq. 5.

Algo. Step 10.1: Determining *common* contextual patterns for candidate relation Steven Gerrard‖England. The following is a list of *all* contextual patterns. The *bold-faced* patterns are common.

– **attacking midfielder**, forward, **have handed**, **player**, **prodigy**, about the signing of, **along with**, are missing, **as striker**, **Goal in**, ace a, **added to**, **and England footballer**, **and England midfielder**, believes, **have confirmed-midfielder**, **winger**

Hence, for candidate relation Steven Gerrard‖England, $\omega_2 = 12$.

Algo. Step 10.2: Determining *common* contextual patterns for candidate relation Tiger Woods‖USA.

– attacking midfielder, forward, **have handed**, **player**, **prodigy**, about the signing of, **along with**, are missing, as striker, Goal in, **ace a**, added to, and England footballer, and England midfielder, **believes**, have confirmed-midfielder, winger

Hence, for candidate relation Tiger Woods‖USA, $\omega_2 = 6$.

Algo. Step 11 The $\text{micro}_{\text{cat}}(h_{ij})$ is calculated as the sum of the two weights ω_1 and ω_2.

– Steven Gerrard‖England, $\text{micro}_{\text{cat}}(h_{ij}) = 4 + 12 = 16$
– Tiger Woods‖USA, $\text{micro}_{\text{cat}}(h_{ij}) = 3 + 6 = 9$

From the above illustration, since Steven Gerrard‖England is ranked much higher, it will be promoted as trusted and added to the trusted set of relations to be used as seed for subsequent iterations.

In summary, the normalization and *S*-function operations determine membership values for contextual patterns of both trusted and candidate relations. This is the fuzzification process based on fuzzy set theory and gives us the graded co-occurrence matrix. The approximation operations based on rough sets are then applied to the fuzzified co-occurrence information.

Table 7 Precision@30 of TPL [29] and FRL for ranking-based method

Categories	TPL			FRL		
	Iter. 1	Iter. 5	Iter.10	Iter. 1	Iter. 5	Iter. 10
Athlete_Team	100	90	87	97	100	97
CEO_Company	100	100	100	100	100	100
City_Country	100	100	100	93	100	100
City_State	100	100	100	97	100	100
Coach_Team	93	93	93	100	100	100
Company_City	83	90	93	97	100	100
Stadium _City	97	93	80	93	70	93
State_Capital	100	97	73	93	83	76
State_Country	100	100	100	90	100	100
Team_vs_Team	93	83	80	100	100	100
Average (%)	96.6	94.6	**90.6**	96	95.3	**96.7**

The upper approximation operator removes all *unrelated* candidates from the set. The lower approximation operator further *prunes* this set. The micro-score obtained by weighting these two rough set operations determines the criteria for promotion of a candidate relation to the trusted set.

5 Experiments

Throughout our experiments, we used the same ontology as in TPL [29] and CBS [39] experiments. We used the same 11 categories as the input ontology: *Athlete_Team, CEO_Company, City_Country, City_State, Coach_Team, Company_City, Stadium _City, State_Capital, State_Country, Team_vs_Team.* We initialized each relational category with 6 seed instances and ran the experiment for 10 iterations. In every iteration, the top 5 new relations in every category were promoted as trusted relations for subsequent iterations.

To facilitate comparison of our FRL algorithm with the other algorithms, we used Precision@N at each iteration. In each iteration, Precision@N is calculated as the ratio of the correct instances to the N-ranked ones. Since the data was not labeled, the correctness of an instance was judged manually.

It took us approximately 76 minutes for each iteration using a Windows 10 machine with 3.40 GHz Intel i7 processor. Table 7 shows the ranking-based Precision@30 results for each category for iterations 1, 5 and 10 for TPL and FRL algorithms. Bold-faced values indicate average values for all categories at the end of iteration 10.

To evaluate promotion-based results, we use the same steps implemented by CPL [4] and TPL [29]. We sampled X pairs from all the promoted pairs and calculated the Precision@30. Table 8 gives the promotion-based results for all three algorithms. In both ranking and promotion-based methods, correctness results were verified manually. Bold-faced values indicate average values for all categories at the end of iteration 10.

Table 8 Precision@30 of TPL [29], CPL [4] and FRL for promotion-based method

Categories	TPL			FRL			CPL
	1	5	10	1	5	10	10
Athlete_Team	100	96	87	100	100	83	100
CEO_Company	100	100	100	100	100	100	100
City_Country	100	100	100	100	93	96	93
City_State	100	100	100	100	100	100	100
Coach_Team	100	100	93	100	100	100	100
Company_City	40	84	97	100	100	100	50
Stadium _City	80	92	70	80	92	90	100
State_Capital	100	100	63	100	88	43	60
State_Country	100	100	100	100	100	100	97
Team_vs_Team	100	84	80	100	96	100	100
Average (%)	92.0	95.6	**89.0**	98.0	96.6	**91.2**	**90.0**

5.1 Analysis of results

Based on the results in Table 7, the average precision after 10 iterations for the proposed FRL algorithm is significantly better than TPL using a ranking-based method. For this dataset, one can observe that FRL is able to handle concept drift better than TPL in all categories. It is important to note that FRL enforces mutual exclusion constraint, whereas TPL was able to maintain high precision with no externally defined constraints. From Table 8, it can be seen that FRL performs better than TPL and CPL. It is noteworthy that CPL enforces 3 forms of constraints during the learning process [4]. Here FRL was able to do better than TPL in terms of concept drift only in two categories, Athlete_Team and Team_vs_Team. With CPL, the only result that was available was after the tenth iteration.

5.2 Handling concept drift in FRL

The concept drift problem is illustrated with examples from Table 9. This table is a snapshot of *promoted* relation instances for all categories after the 10th iteration. Misclassified relations are shown in bold. Notice that relation Israelites_Joshua has been incorrectly promoted to *category* Athlete_Team. When the dataset is limited, after N iterations, candidates from different categories are misidentified as trusted. The most egregious case in this table is for the category State_Capital City which includes several misclassifications. For example, India_Mumbai was promoted to State_Capital City category when in fact, it should have been promoted to City_Country category. In Table 9, for category City_Country, there are no misclassifications even after 10 iterations. The reason for these scenarios is that, in our dataset, there were sufficient instances for category City_Country and insufficient instances for category State_Capital City. Hence, concept drift is also an outcome of the sample size.

To overcome this problem, FRL enforces mutual exclusion by calculating a mutex score for newly trusted relation with the top ranked relation for that iteration using Eq. 8.

$$\text{mutex} = \frac{\omega_{h_{uv}}}{\omega_{h_{ij}}} \times 100 \tag{8}$$

Table 9 Promoted *relation* instances after 10th iteration for all categories

Relations	Categories
Gators‖Tebow, Colts‖Peyton Manning, Ben Roethlisberger‖Steelers, Eli Manning‖Giants, **Israelites**‖**Joshua**	Athlete_Team
Chris Rudge‖COC, Rex Tillerson‖Exxon, John Alexander‖PBL, Richard Scudamore‖Premier League, Nick Fry‖Honda Racing	CEO_Company
New Zealand‖Wellington, Budapest‖Hungary, Stockholm‖Sweden, Finland‖Helsinki, Amsterdam‖Netherlands	City_Country
Lincoln‖Nebraska, Oklahoma‖Oklahoma City, Columbia‖Missouri, Jackson‖Mississippi, Charlotte‖North Carolina	City_State
Harry Redknapp‖Portsmouth, Atletico‖Javier Aguirre, France‖Raymond Domenech, Juande Ramos‖Tottenham, Luis Aragones‖Spain	Coach_Team
Gaza‖UN, Baghdad‖United Nations, CIA‖Langley, IBM‖New York, Ottawa‖RCMP	Company_HQ City
New York‖Yankee, Indianapolis‖Lucas Oil, Omaha‖Rosenblatt, Dodger‖Los Angeles, **East Rutherford**‖**Giants**	Stadium_City
India‖**Mumbai**, **Delaware**‖**Wilmington**, **Canada**‖**Ontario**, **Long Island**‖**New York**, **Delhi**‖**India**	State_Capital City
South Carolina‖United States, Mississippi‖United States, Iowa‖United States, New Hampshire‖United States, United States‖West Virginia	State_Country
Liverpool‖Manchester United, Packers‖Vikings, Knicks‖Lakers, Patriots‖Steelers, Barcelona‖Real Madrid	Team_vs_Team

Table 10 Relations potentially belonging to more than one category

Relations	Categories
Apple‖Jobs	**CEO_Company**, Athlete_Team
New York‖United States	**City_Country**, City_State
Apple‖SteveJobs, BillGates‖Microsoft, Microsoft‖Steve Ballmer	**CEO_Company**, Coach_Team
Colorado‖Denver, Arizona‖Phoenix, Atlanta‖Georgia, Manitoba‖Winnipeg, Madison‖Wisconsin	**State_Capital**, City_State
Toronto‖Ontario, Alberta‖Calgary, Boston‖Massachusetts	State_Capital, **City_State**

where $\omega_{h_{ij}}$ and $\omega_{h_{uv}}$ are the calculated micro-scores for relations to be parsed and top ranked trusted relations, respectively.

Example 1 The initial seed for the *category* State_Capital included the following set of relations:

- Manitoba‖Winnipeg, Alabama‖Montgomery, California ‖Sacramento
- Florida‖Tallahassee, Georgia‖Atlanta, Minnesota ‖Saint Paul

Note, that all samples are equally qualified as trusted instances for *category* City_State, but are mutually exclusive to State_Capital and hence if promoted in any other category as trusted were ignored.

In Table 10, we give some examples of relations that had the potential to be promoted as *trusted* in two separate categories. However, these relations were ignored as their mutex score was higher for the one category and hence excluded from the others. Categories shown in **bold** had a higher mutex score; therefore, the associated relation was mutually exclusive to that category.

Table 11 Precision@30 of TPL [29], CBS [39] and **FRL** [2] for **nouns**

Categories	Iteration 5			Iteration 10		
	TPL (%)	CBS (%)	FRL (%)	TPL (%)	CBS (%)	FRL (%)
Company	100	100	100	100	100	100
Disease	100	100	100	100	100	100
KitchenItem	100	94	97	100	94	73
Person	100	100	100	100	100	100
PhysicsTerm	93	100	67	90	100	77
Plant	100	100	77	97	100	100
Profession	100	100	100	100	87	100
Sociopolitics	100	48	93	100	34	87
Sport	97	97	100	100	100	100
Website	90	94	97	90	90	93
Vegetable	93	83	83	63	48	47
Average	**97.5**	**92**	**92**	**94.5**	**87**	**89**

5.3 FRL for *nouns*

To complete the discussion of both linguistic entities (nouns and relations), we have included the results from [2]. Here, the dataset includes 68,919 noun phrase instances and 59,325 contextual patterns for nouns. The experimental set up was exactly the same as in TPL [29] and CBS [39] experiments. We used the same 11 categories as the input ontology: *Company, Disease, KitchenItem, Person, PhysicsTerm, Plant, Profession, Sociopolitics, Website, Vegetable, Sport.*

We initialized each category with 5–6 seed instances and ran the experiment for 10 iterations. In every iteration, the top 5 new noun phrases for every category were promoted as trusted nouns for subsequent iterations. Table 11 shows the result for all categories for Precision@30. One can observe that FRL algorithm performs better than CBS but not as good as TPL in terms of average precision value over all 11 categories and 10 iterations. In the case of *nouns*, the FRL algorithm does not enforce any mutual exclusion constraints. This was due to lack of richness of the dataset. For example, the category *vegetables* did not have sufficient seeds, and as a result, all three algorithms fared poorly with this category. Here, parameter values α and β for S-function were identical to that of the relations. Bold-faced values indicate average values for all categories.

6 Complexity and scalability issues

The complexity of FRL is affected by several individual parameters. In FRL, step 1 has a linear time and space complexity $O(\mathcal{H})$, where $\mathcal{H}$ is the number of relations. Step 2 has a complexity $O(\#_{\text{cat}})$ where each $\#_{\text{cat}}$ represents number for categories. Steps 3, 4 and 8 require $O(|\text{CO}|)$ to form fuzzy sets where CO represents mapping for each relation in $\mathcal{R}$ with the co-occurring context, and hence, the complexity is $O(|R|)$ where R is the number of contexts for each relation. Steps 5 to 7 and 9 to 11 are a linear pass, and repeat steps 4 and 8 have a complexity of $O(|R|)$, respectively.

Considering that most of the factors remain consistent through all of the iterations, the most expensive or time consuming part of FRL is to calculate the fuzzy thesaurus as well as the upper and lower approximations. Thus, the time complexity of the algorithm is $O(\mathcal{H})$ for each candidate relation. For space complexity, we preprocess to filter out low cardinality co-occurring values and hence the complexity is $O(|\mathcal{H}| \times |\mathcal{R}|)$ based on the number of co-occurring nonzero values.

In terms of scalability, FRL was tested on single thread CPU in a non-continuous (unchanged dataset) environment. Most of the tasks such as calculating fuzzy memberships and upper and lower approximations for each category can be processed in parallel with GPU computing. However, for a continuous ever-changing dataset environment, some sort of pre-processing of initial data needs to be done. Also, scalability depends on (i) learning nouns vs. relations (noun pairs), (ii) the richness of the examples, (iii) the number of categories, and (iv) quality of linguistic entities within each categories. The last factor has a direct bearing on the number of constraints that need to be defined to handle concept drift.

7 Conclusion

We have proposed a novel semi-supervised learning algorithm (FRL) based on fuzzy rough sets for labeling relations using contextual pattern information. We introduced a formal framework of the fuzzy rough set for linguistic entities. The proposed FRL algorithm was experimentally compared with a tolerance rough set-based learner (TPL) and the coupled pattern learner (CPL). The choice of methods for comparative study was motivated by the fact that fuzzy rough sets and tolerance rough sets permit overlapping (or soft similarity) of classes. CPL and CBS (for nouns) methods were used as benchmarks for this work, and hence, the same datasets were used in all the experiments. Experimental results demonstrate that for this dataset, FRL (for relations) performs better than TPL based on the ranking method and outperforms CPL and TPL using the promotion-based method. In terms of representation, with the tolerance rough set model, a *crisp thesaurus* is constructed using a tolerance value ε. Determination of the optimal value of ε is a challenge. With the fuzzy rough set model, the challenge was to determine the optimal parameters α and β for the S-function to construct a *graded thesaurus*. For handling concept drift, this work also reveals that it was necessary to define mutual exclusion constraints for FRL. CPL defines several constraints, and TPL requires no constraints. This could be important since constraints can add to the computational overhead as the number of categories, and the size datasets grow in an ever learning environment typically associated with web corpora. As future work, we plan to explore the capabilities of FRL and TPL over more categories and larger datasets for categorizing nouns and relations.

Acknowledgements Special thanks to Cenker Sengoz for sharing the dataset and for discussions regarding TPL. We are very grateful to Prof. Estevam R. Hruschka Jr. for the NELL dataset and Prof. Andrzej Skowron for helpful suggestions.

References

1. Banko M, Cafarella M, Soderland S, Broadhead M, Etzioni O (2007) Open information extraction from the web. In: Proceedings of IJCAI, pp 2670–2676
2. Bharadwaj A, Ramanna S (2017) Fuzzy rough set-based unstructured text categorization. In: Mouhoub M, Langlais P (eds) Canadian AI 2017, LNAI 10233, pp 335–340

3. Brin S (1999) Extracting patterns and relations from the world wide web. In: Selected papers from the international workshop on the world wide web and databases, WebDB'98, pp 172–183
4. Carlson A, Betteridge J, Wang RC, Hruschka Jr ER, Mitchell TM (2010) Coupled semi-supervised learning for information extraction. In: Proceedings of the 3rd ACM international conference on web search and data mining, pp 101–110
5. Cock MD, Cornelis C, Kerre EE (2004) Fuzzy rough sets: beyond the obvious. In: Proceedings of the 2004 IEEE international conference on fuzzy systems, vol 1, pp 103–108
6. Cornelis C, De Cock M, Radzikowska AM (2008) Fuzzy rough sets: from theory into practice. In: Pedrycz W, Skowron A, Kreinovich V (eds) Handbook of granular computing. Wiley, Hoboken, pp 533–552
7. Curran J, Murphy T, Scholz B (2007) Minimising semantic drift with mutual exclusion bootstrapping. In: Proc. of PACLING, pp 172–180
8. De Cock M, Cornelis C (2005) Fuzzy rough set based web query expansion. In: Proceedings of rough sets and soft computing in intelligent agent and web technology, pp 9–16
9. Dong XL, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, Strohmann T, Sun S, Zhang W (2014) Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: The 20th ACM SIGKDD international conference on knowledge discovery and data mining, KDD'14, New York, pp 601–610
10. Dubois D, Prade H (1990) Rough fuzzy sets and fuzzy rough sets*. Int J Gener Syst 17(2–3):191–209
11. Etzioni O, Fader A, Christensen J, Soderland S, Mausam (2011) Open information extraction: the second generation. In: International joint conference on artificial intelligence, pp 3–10
12. Gama J, Zliobaite I, Bifet A, Pechenizkiy M, Bouchachia A (2013) A survey on concept drift adaptation. ACM Comput Surv 1(1):1–44
13. Ghahramani Z, Heller KA (2005) Bayesian sets. In: Advances in neural information processing systems, vol 18
14. Ho TB, Nguyen NB (2002) Nonhierarchical document clustering based on a tolerance rough set model. Int J Intell Syst 17:199–212
15. Jensen R, Shen Q (2008) Computational intelligence and feature selection: rough and fuzzy approaches, vol 8. Wiley, London
16. Kawasaki S, Nguyen NB, Ho TB (2000) Hierarchical document clustering based on tolerance rough set model. In: Proceedings of the 4th European conference on principles of data mining and knowledge discovery, pp 458–463
17. Mahdisoltani F, Biega J, Suchanek FM (2015) YAGO3: a knowledge base from multilingual wikipedias. In: 7th Biennial conference on innovative data systems research (CIDR 2015)
18. Miller GA (1995) Wordnet: a lexical database for english. Commun ACM 38(22):39–41
19. Mitchell T, Cohen W, Hruschka E, Talukdar P, Betteridge J, Carlson A, Dalvi B, Gardner M, Kisiel B, Krishnamurthy J, Lao N, Mazaitis K, Mohamed T, Nakashole N, Platanios E, Ritter A, Samadi M, Settles B, Wang R, Wijaya D, Gupta A, Chen X, Saparov A, Greaves M, Welling J (2018) Never-ending learning. Commun ACM 61(5):103–115
20. Ngo CL (2003) A tolerance rough set approach to clustering web search results. Master's thesis, Warsaw University
21. Nguyen H, Ho TB (2008) Rough document clustering and the internet. In: Pedrycz W, Skowron A, Kreinovich V (eds) Handbook of granular computing. Wiley, Hoboken, pp 987–1003
22. Nguyen S, Swieboda W, Jaskiewicz G (2012) Extended document representation for search result clustering. In: Bembenik R, Skonieczny L, Rybinski H, Niezgodka M (eds) Intelligent tools for building a scient. Info. Plat. SCI, vol 390, pp 77–95
23. Pal SK, Skowron A (eds) (1999) Rough-fuzzy hybridization: a new trend in decision making, 1st edn. Springer, Secaucus
24. Pawlak Z (1982) Rough sets. Int J Comput Inf Sci 11(5):341–356
25. Polkowski L, Skowron A, Zytkow J (1994) Tolerance based rough sets. In: Lin TY, Wildberger M (eds) Soft computing: rough sets, fuzzy logic, neural networks, uncertainty management, knowledge discovery. Simulation Councils Inc., San Diego, pp 55–58
26. Radzikowska AM, Kerre EE (2002) A comparative study of fuzzy rough sets. Fuzzy Sets Syst 126:137–156
27. Ramanna S, Peters J, Sengoz C (2017) Application of tolerance rough sets in structured and unstructured text categorization: a survey. In: Wang G (ed) Thriving rough sets, studies in computational intelligence, vol 708. Springer, Cham, pp 119–137
28. Rebele T, Suchanek F, Hoffart J, Biega J, Kuzey E, Weikum G (2016) YAGO: a multilingual knowledge base from wikipedia, wordnet, and geonames. Springer, Cham, pp 177–185
29. Sengoz C (2014) A granular-based approach for semi-supervised web information labeling. Master's thesis, University of Winnipeg

30. Sengoz C, Ramanna S (2014) A semi-supervised learning algorithm for web information extraction with tolerance rough sets. In: Active media technology 2014, Web Intelligence Conference 2014, LNCS 8610, pp 1–10
31. Sengoz C, Ramanna S (2015) Learning relational facts from the web: a tolerance rough set approach. Pattern Recogn Lett 67(P2):130–137
32. Shi L, Ma X, Xi L, Duan Q, Zhao J (2011) Rough set and ensemble learning based semi-supervised algorithm for text classification. Expert Syst Appl 38(5):6300–6306
33. Skowron A, Stepaniuk J (1996) Tolerance approximation spaces. Fundam Inf 27(2,3):245–253
34. Srinivasan P, Ruiz ME, Kraft DH, Chen J (2001) Vocabulary mining for information retrieval: rough sets and fuzzy sets. Inf Process Manag 37(1):15–38
35. Suchanek FM (2009) Automated construction and growth of a large ontology. PhD thesis, Natural Sciences and Technology of Saarland University
36. Suchanek FM, Kasneci G, Weikum G (2007) Yago: a core of semantic knowledge. In: 16th international world wide web conference (WWW 2007). ACM Press, New York, pp 697–706
37. Swieboda W, Meina M, Nguyen H (2013) Weight learning for document tolerance rough set model. In: RSKT 2013, LNAI 8171. Springer, Berlin, pp 386–396
38. Thanh NC, Yamada K, Unehara M (2011) A similarity rough set model for document representation and document clustering. J Adv Comput Intell Intell Inf 15(2):125–133
39. Verma S, Hruschka Jr ER (2012) Coupled Bayesian sets algorithm for semi-supervised learning and information extraction. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 307–322
40. Virginia G, Nguyen HS (2013) Lexicon-based document representation. Fundam Inf 124(1–2):27–46
41. Virginia G, Nguyen HS (2015) A semantic text retrieval for indonesian using tolerance rough sets models. Trans Rough Sets LNCS 8988(XIX):138–224
42. Zadeh L (1997) Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets Syst 177(19):111–127

Aditya Bharadwaj is an Application Support Analyst working with Great West Life Assurance Company. Aditya received his M.Sc. degree in Applied Computer Science from University of Winnipeg, Canada, and BE degree in Information Technology from RTM Nagpur University, India. Aditya's co-authored paper on Fuzzy Rough Set-based text categorization appeared in the Proceedings of the Canadian AI 2017. He was the recipient of Graduate Scholarship at University of Winnipeg. His areas of interest include: semi-supervised learning, data analytics, and text categorization and its applications.

Sheela Ramanna is a Full Professor, past Head and Graduate Program Cofounder of the Applied Computer Science Department at University of Winnipeg, Canada. She received a Ph.D. (CS) from KSU, USA, BS (EE) and MS (CS) from Osmania University, India. She is on the EB of the Springer TRS Journal, IJKESDP Journal, Assoc. Editor of KES Journal, and Senior Member of the IRSS. She has co-edited a book on Emerging Paradigms in Machine Learning, Springer, 2013. She served as Program Co-Chair for MIWAI 2013, RSKT 2011, RSCTC 2010 and JRS2007. She has published over 40 articles in the past 6 years. She is the recipient of 2015 TUBITAK Fellowship (Turkey). She has received more than $1,118,000 in research funding since 1992. Her major research includes foundations of fuzzy, rough and near sets with applications in social networks, text categorization, perception-based image and audio data, and forms of descriptive proximities and its applications.