Unsupervised Discovery of Relations and Discriminative Extraction Patterns

Alan Akbik Larysa Visengeriyeva
Priska Herger Holmer Hemsen Alexander Löser
Technische Univeristät Berlin
Database Systems and Information Management Group
Einsteinufer 17, 10587 Berlin, Germany
`{firstname.lastname}@tu-berlin.de`

ABSTRACT
Unsupervised Relation Extraction (URE) is the task of extracting relations of *a priori* unknown semantic types using clustering methods on a vector space model of entity pairs and patterns. In this paper, we show that an informed feature generation technique based on dependency trees significantly improves clustering quality, as measured by the F-score, and therefore the ability of the URE method to discover relations in text. Furthermore, we extend URE to produce a set of weighted patterns for each identified relation that can be used by an information extraction system to find further instances of this relation. Each pattern is assigned to one or multiple relations with different confidence strengths, indicating how reliably a pattern evokes a relation, using the theory of Discriminative Category Matching. We evaluate our findings in two tasks against strong baselines and show significant improvements both in relation discovery and information extraction.

KEYWORDS: Unsupervised Relation Extraction, Clustering, Vector Space Models.

1 Introduction

Recently, there has been great interest in broadening information extraction methods to allow for unsupervised discovery of relational information in large document collections of unknown content. Contrary to classic information extraction in which relationship types (such as BornIn or MarriedTo) are specified in advance, such methods automatically identify *a priori* unknown relationship types in a given corpus. For these identified semantic relations[1], they subsequently or simultaneously perform an information extraction step, thereby transforming the corpus into structured, relational data without any supervision or previous knowledge about its content.

One such approach, Unsupervised Relation Extraction (URE), addresses this challenge by building on the *latent relation hypothesis* which states that pairs of words that co-occur in similar patterns tend to have similar relations (Turney, 2008; Rosenfeld and Feldman, 2007). Current techniques capture this in a vector space model by computing a *pair-pattern matrix* in which each row represents an entity pair and each column a distinct pattern, with co-occurrence counts as cell values. This representation allows us to compute the similarity of two entity pairs by comparing the distribution over observed patterns. Using such a similarity metric, clustering methods can find clusters of entity pairs that share similar patterns and can therefore be assumed to represent a relation. Ideally, a clustering method returns three kinds of structured information, each of which is highly relevant to information discovery and extraction in unknown corpora: Firstly, a set of clusters, each of which represents one distinct *relation*. Secondly, for each cluster a set of *entity pairs* between which this relation holds. Thirdly, for each cluster a set of discriminative *patterns* that extensionally describe the relation and may be used by an information extraction system to find further *relation instances* of this relation.

The choice of patterns as well as their significance within a cluster assignment are crucial aspects to the success of this endeavor. Any one pattern may be underspecified or ambiguous and give different amounts of explicit or implicit evidence to different relations. Worse, as the complexity of language permits for one relation to be expressed in a multitude of ways, we may expect the distribution of patterns observed for each relation to be heavy-tailed, with a few patterns observed in high numbers and a large number of very rare patterns.

Take, for instance, the relation MarriedTo: Patterns that indicate this relation range from explicit and discriminative expressions, such as "*X married Y*" and "*X married to Y*", over entailment, such as "*X divorced from Y*" and "*X ex-wife of Y*", to mere implicit evidence, such as "*X fell in love with Y*". Here, *X* and *Y* are placeholders for an entity each. At the same time, these patterns may also express other relations at varying degrees; the pattern "*X divorced from Y*" for example explicitly expresses the DivorcedFrom relation, while also entailing the MarriedTo relation. The desired result should reflect this and allow one-to-many assignments of patterns to relations, in which each pattern-relation assignment is weighted according to a *distinctiveness* value: High distinctiveness indicates a pattern that explicitly and unambiguously evokes a relation, low distinctiveness more implicit or ambiguous patterns.

In this paper, we examine more closely the task of discovering and ranking discriminative patterns for each relation and the impact of the choice of pattern generation scheme on overall URE results. By focusing on patterns, URE benefits in two ways: On the one hand we show that an informed feature generation strategy can markedly reduce the amount of underspecified and ambiguous patterns in the pair-pattern matrix, thereby significantly improving clustering

[1] In this paper, we refer to relationship types as *relations* and to instances of relationship types as *relation instances*.

approaches. On the other hand, this allows us to extend URE to not only identifying relations, but also finding and ranking a list of patterns for each relation that can be used in subsequent information extraction.

Contributions. We propose an unsupervised approach that identifies relations in a corpus of unknown content by clustering entity pairs and characterizes each relation by finding a list of patterns ordered according to the amount of explicit evidence they give to the presence of the identified relation. The contributions of this paper can be summarized as follows:

Algorithm for feature selection in a dependency graph. We propose an algorithm that selects possible patterns for a given entity pair in a dependency path, as an extension of the *shortest path* method. The approach is capable of capturing a wider range of phenomena than previous part-of-speech based feature generation and filtering approaches by incorporating syntactic elements for long range dependencies, complements for light or support verbs, appositions and context for arguments in direct conjunction. We show that the proposed feature selection technique increases the clustering quality F-measure by 65% over baseline approaches and that identified patterns are better suited to be used in an information extraction task.

Method for computing weighted pattern-relation assignments. We propose an approach that uses clustering results to compile a set of pattern-relation assignments, weighted according to the amount of discriminative evidence each pattern gives to an assigned relation. The method is based on the theory of Discriminative Category matching (Fung et al., 2002). We experimentally show that these assignments produce patterns suitable for the task of information extraction.

We evaluate the proposed method in two different tasks: A *clustering task* in which we evaluate our clustering approach on three ground truth datasets of different composition against three baseline approaches [2]. We investigate the impact of our proposed feature selection algorithm on overall clustering quality and its ability to find the optimal amount of relationships in different datasets. Secondly, an *information extraction task* in which we evaluate the ranked patterns on two gold standard corpora and compare precision and recall with a baseline approach.

The remainder of the paper is organized as follows: Section 2 reviews previous work in the area of clustering for unsupervised relation discovery. We outline several approaches used as baselines in our evaluation. Section 3 outlines our clustering approach, illustrating in Section 3.1 our proposed algorithm for pattern extraction in dependency trees, and in Section 3.3 our proposed method for identifying and ranking discriminative patterns. Section 4 describes evaluation methods, experimental setup, datasets and reports the results on the two evaluation tasks. Finally, Section 5 concludes this paper.

2 Related Work

Most previous work utilizes the pair-pattern matrix to either measure the similarity of pairs of words, or to measure the similarity between patterns for a number of different purposes. In this section we review this work with respect to our pattern extraction and clustering approach and identify evaluation baselines.

[2] The datasets used in our experiments are available on request for research purposes.

Relation discovery. (Rosenfeld and Feldman, 2007) cluster entity pairs in the pair-pattern matrix to identify semantic relations. The resulting clusters are interpreted as each representing one relation that holds between all entity pairs in the cluster. They use the text between two entities in a sentence as patterns, but also allow arbitrary word skips, meaning that for each sentence containing an entity pair a large number of features are generated. They cluster the matrix using k-means and hierarchical agglomerative approaches and find that better results are reached with a complex feature space. (Bollegala et al., 2010) propose a co-clustering approach that simultaneously clusters both entity pairs and patterns for identifying relations, using not only lexical, but also shallow syntactic patterns. They expand the feature set to also include prefix and postfix spans. More recently, (Wang et al., 2011) analyzed the impact of filtering techniques and found that overall clustering quality F-measure significantly increases by using a set of filters to eliminate patterns that are unlikely to represent a relation. They filter out a total of 80% of all observed patterns. They use the text between entities as patterns, without word skips, and include named entity class information into the feature set.

Contrary to previous approaches in relation discovery, we employ a feature generation technique that utilizes information from a dependency parser. Our observation is that current dependency parsers are becoming orders of magnitudes faster while retaining a sufficiently high precision and recall (see (Rush and Petrov, 2012) and (Zhang and Nivre, 2011)). We comparatively evaluate our feature generation technique against baselines modeled after the three approaches mentioned above.

Similarity of patterns or words. Instead of using clustering to identify relations, much work has focused on measuring the pairwise similarity of patterns or words. (Turney, 2006) compute the pairwise similarity of lexical patterns to solve the problem of finding analogies between word pairs. (Turney, 2011) compare pairs of words using the distribution over patterns to find proportional analogies and evaluate this on corpora of word comprehension tests, such as analogy questions in SAT or TOEFL tests. By contrast, (Lin and Pantel, 2001) directly measure the pairwise similarity between patterns in dependency trees using the distribution over word pairs to find inference rules from text. (Sun and Grishman, 2010) extend this with a clustering approach to group patterns into clusters, which they use to guide semi-supervised relation extraction methods. While this approach returns clusters of patterns for each discovered relation, the clustering is “hard”, meaning that each pattern is assigned to exactly one cluster. This is contrary to our intuition that each pattern may give different amounts of evidence to different semantic relations. Nevertheless, we use a reimplementation of this approach as baseline for the evaluation of our proposed pattern ranking method.

3 Relation Discovery and Pattern Ranking

We propose a method that takes as input a document collection, identifies relations by clustering entity pairs with a similar pattern distribution, and outputs a ranked list of patterns for each relation. This is done in three steps: First, we generate the pair-pattern matrix using a feature generation approach based on deep syntactic analysis as explained in Section 3.1. Second, we run a clustering algorithm to group entity pairs into clusters representing relations (see Section 3.2). Finally, we compute the distinctiveness for each pattern in each cluster based on the distribution of patterns both within and across generated clusters as detailed in Section 3.3.

3.1 Feature Generation Using Dependency Trees

The proposed feature generation algorithm takes as input a set of dependency parsed sentences and entity pairs[3]. For each sentence and entity pair it generates a list of patterns that are used as features for the entity pair. The method determines a set of *core tokens* by collecting all tokens on the shortest path between the two entities. It then finds a set of *optional tokens* by collecting all tokens linked to a core token with certain typed dependency. It generates one feature for each combination of the core tokens and the power set (the set of all possible subsets) of the optional tokens.

Typed dependencies that indicate possibly important information even if not on the shortest path were determined through experimentation. Simple examples of cases in which important information is not on the shortest path are negation and particles, which are directly connected to a verb (with the dependencies "*neg*" and "*prt*" respectively) but never function as a link on the path between two arguments bound by this verb. Other examples are appositions, which may be connected to an entity but are not themselves part of the shortest path (indicated by "*nn*" or "*appos*"), and light verb constructions in which only the verb, but not the typically more important noun is part of the shortest path. Another example - discussed in detail below - are two entities in conjunction that function as an argument for a verb.

The method consists of four steps:

Step 1: Compute the shortest path between subject and object. The shortest path between two entities in a dependency path serves as basis for our extraction method. Recent research shows that lexical tokens along the shortest path represent particularly discriminative patterns for extraction of binary and even higher-order relations (Etzioni et al., 2011; Akbik and Broß, 2009). By focusing on the tokens that syntactically link both entities, we can skip over tokens that are less likely to be relevant to the relationship. This step yields a list of core tokens likely to be relevant to the relation expressed between the two entities.

Step 2: Collect of a set optional tokens on the path. We collect all tokens that *may* be relevant to identifying a relation by iterating over each token on the shortest path and examining all typed dependencies of each token to non-path tokens. If the dependency is one of {*nn*, *neg*, *prt*, *poss*, *possessive*, *nsubj*, *nsubjpass*} we collect the target token into a list of optional tokens. This step yields a list of tokens to be added to the core list to produce a good extraction pattern.

Step 3: Generate features. We build the power set over all optional tokens and generate one feature for each combination of the shortest path and optional set. This power set includes the empty set as well, so the shortest path without any optional tokens is included in the features.

Step 4: Remove uninformative features. We filter all features that consist only of closed-world word classes. Examples are features like "*X and Y*" or "*X of Y*". The intuition for this step is that such patterns are semantically too weak to be used as patterns and not suitable for clustering approaches.

The following example sentence illustrates the feature generation process: *"James Joyce and his longtime lover Nora Barnacle got married in 1931*". Figure 1 depicts the sentence's dependency parse. Here, the shortest path is a "*conj*"-link, directly connecting the two entities "*James Joyce*"

[3] We use the Stanford dependency parser (Klein and Manning, 2003) and Stanford typed dependencies (De Marneffe et al., 2006) in our experiments

and ”*Nora Barnacle*“. The resulting pattern ”*X and Y*“[4] is highly ambiguous and therefore of limited use. We collect the tokens ”*and*“, ”*his*“, ”*lover*“ and ”*married*“ into a set of optional tokens and build its power set. By taking each combination of the power set and the shortest path (and after filtering non-informative features) we arrive at a total of five features. Table 1 lists them and compares them to shallow patterns as generated by (Turney, 2011) and (Wang et al., 2011).

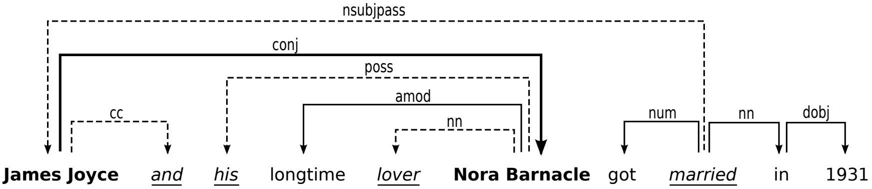

Figure 1: Dependency parse of the example sentence. The entity pair and shortest path are marked in bold. ”*James Joyce*“ and ”*Nora Barnacle*“ are directly connected with a ”*conj*“ link. Links to optional tokens are illustrated as dotted lines; optional tokens are underlined.

(Turney, 2011)	**X** and his longtime lover **Y**, **X** and his longtime * **Y**, **X** and his * lover **Y**, **X** and * longtime lover **Y**, **X** * his longtime lover **Y**, [..]
(Wang et al., 2011)	**PERSON** and his longtime lover **PERSON**
PROPOSED	**X** and lover **Y**, **X** and **Y** married, **X** and lover **Y** married, **X** and his lover **Y**, **X** and his lover **Y** married

Table 1: Features from different generation methods for the sentence in Figure 1. We observe that the features generated by the proposed approach all indicate the MarriedTo relation either explicitly or implicitly. By contrast, the shallow feature generation technique used by (Turney, 2011) produces a total of 24 patterns, many of which are highly underspecified. (Wang et al., 2011) generates only one, overspecified feature.

3.2 Relation Discovery by Clustering Entity Pairs

From the features as generated according to Section 3.1 we build a pair-pattern matrix for all entity pairs observed at least 20 times in the corpus. This is accomplished by counting co-occurrences between patterns (features) and entity pairs. We cluster this vector space model using the k-means algorithm which partitions entity pairs into k clusters of similar variance. Following (Bullinaria and Levy, 2007), we use the Cosine similarity to measure distances between feature vectors – a measure useful for highly sparse vectors like the ones at hand. The k-means algorithm requires us to manually specify the number of output clusters which

[4] In this case, the pattern ”X and Y“ is a verbalization of the entities X and Y being linked by the typed dependency ”*conj*“ for readability reasons.

in turn allows us to control the granularity of discovered relations. For instance, the cluster representing the relation ChildOf for low values of k is split into two clusters representing the relations SonOf and DaughterOf given higher values of k. Following (Rosenfeld and Feldman, 2007) we interpret each cluster within a clustering as a distinct, unlabeled relation and all entity pairs as relation instances.

3.3 Ranking Patterns by Distinctiveness

A clustering assigns each entity pair to a cluster and thereby implicitly produces a set of patterns per cluster, namely all non-zero features of the entity pairs in that cluster. The approach proposed here for ranking these patterns is based on two intuitions. The first being that clusters are representative of relations, meaning that the distribution of patterns in a cluster dominantly contains a single relation. This implies that patterns that are shared by a majority of entity pairs in a cluster are common ways of expressing a relation, while patterns that are shared only by few entity pairs are either less commonly used or provide only implicit evidence of a relation. To capture this intuition we compute pattern weights by adding up the counts per pattern over all entity pairs within a cluster. We normalize the value to compute the *significance* of a pattern:

$$Significance_{i,R} = \frac{\log_2(f_{i,R}+1)}{\log_2(P_R+1)} \tag{1}$$

As the equation shows, the significance of a pattern i within a cluster R is denoted as the logarithmic ratio of its weight $f_{i,R}$ normalized by the sum over all pattern weights P_R in the cluster.

The second intuition is that patterns that occur in more than one cluster may be ambiguous and lend different amounts of evidence to different relations. Such patterns therefore have low *clarity*. We collect the distributed evidence of these patterns across clusters and relate it to a pattern's highest significance over all clusters. The following equation measures this clarity for each pattern, with $0 \leq Clarity_i \leq 1$.

$$Clarity_i = \begin{cases} \log_2 \dfrac{n \cdot \max\limits_{j\in\{1..n\}} \left\{Significance_{i,R_j}\right\}}{\sum\limits_{j=1}^{n} Significance_{i,R_j}} \cdot \dfrac{1}{\log_2 n}, & n > 1 \\ 1, & n = 1 \end{cases} \tag{2}$$

Thereby, a pattern i has a high $Clarity_i$ if it is significant in one cluster and insignificant in the others. (Note that $1/\log_2 n$ is a normalization factor.) If we observe a pattern only once and in one cluster its $Clarity$ is 1.

Following the theory of Discriminative Category Matching (DCM) (Fung et al., 2002), the overall *distinctiveness* of a pattern given a cluster is a combination of *Significance* and *Clarity* accordingly; with a normalization factor of $\sqrt{2}$:

$$Distinctiveness_{i,R} = \frac{Significance_{i,R}^2 \cdot Clarity_i^2}{\sqrt{Significance_{i,R}^2 + Clarity_i^2}} \cdot \sqrt{2} \tag{3}$$

We use this $Distinctiveness$ measure to re-weigh pattern-cluster assignments and produce a ranked list of patterns for each cluster.

4 Evaluation

We quantify the proposed approach in two tasks: a *clustering task* to measure the impact of our feature generation method on overall clustering performance and evaluate the ability to discover relations. And an *information extraction task* to examine clustered patterns with respect to their usefulness to information extraction. Since ground truth is not usually readily available for large amounts of text, assessing the quality of large scale clustering results has proven to be difficult. We therefore use distant supervision based on the YAGO knowledge base to automatically construct various ground truth data sets. Details of the set-up, advantages and drawbacks of such an evaluation approach as well as measures used to analyze clustering quality against such ground truth are discussed in Section 4.1. Details on clustering evaluation and the information extraction task are discussed in Sections 4.2 and 4.3 respectively.

4.1 Experimental Setup

4.1.1 Datasets

The extrinsic evaluation of URE is problematic as it requires a document collection with exact knowledge regarding its content in the form of labeled relation triples. Several projects have constructed such a ground truth manually, which has a number of drawbacks: Firstly, there is a high cost involved in manually annotating sentences with relations, limiting the size of the ground truth as well as the ability to quickly generate new evaluation sets. Secondly, much care must be taken to ensure that no URE-specific assumptions are modeled into the ground truth, i.e. "overfitting" the ground truth to the capabilities of the algorithm that is to be evaluated. The inherent risk in manual annotation is the creation of a ground truth that does not realistically reflect the application scenario the URE approach is intended for.

We therefore choose a *distant supervision*-based approach to automatically generate a number of labeled training, test and evaluation sets. In distant supervision, an existing knowledge base of facts (triples consisting of two entities and a relation that holds between the entities) is used as support tool (Mintz et al., 2009). We use YAGO, a semantic knowledge base derived from Wikipedia, WordNet and GeoNames with knowledge of more than 10 million entities and around 447 million facts (Hoffart et al., 2011). The relations in YAGO are semantic labels, such as WasBornIn, ActedIn and DiedIn and are therefore different from a textual representation of these relations in a sentence.

The approach randomly selects a number of entity pairs from the knowledge base and retrieves from the Web[5] a set of sentences containing each entity pair. The assumption is that a sentence that contains an entity pair for which the knowledge base specifies a relation is likely to express it, either explicitly or implicitly. Accordingly, this allows the method to automatically label all retrieved sentences with relations, enabling the generation of a ground truth of arbitrary size. In order to assess the quality of the ground truth we manually examine 200 sentences with a total of 209 relations and 29 distinct relations [6]. We find that in 159 cases the relation is either explicitly or implicitly represented in the sentence, whereas in 50 cases the entity pair is present in the sentence but the YAGO relation between them could not be inferred from the text.

Examples for explicit, implicit and false sentences are given in Table 2. While imperfect, the

[5] Using the Bing API (*http://www.bing.com/developers/*).

[6] In 9 cases, one entity pair has more than one relation in YAGO. An example are persons that both ActedIn and Produced a movie.

assumption therefore holds for approximately 76% of the generated ground truth. For our evaluation purposes we find this satisfactory, as this realistically simulates noise while reliably indicating the relational content of generated evaluation sets.

sentence retrieved	relation expressed
Mystery Men *(1999) stars* ***Ben Stiller*** *as Mr. Furious.*	explicit
Mystery Men *brought on board a talented cast from William H. Macy to* ***Ben Stiller****.*	explicit
What was ***Ben Stiller****'s character's super quality in* ***Mystery Men****?*	implicit
Ben Stiller *does not think* ***Mystery Men*** *should be remade.*	false

Table 2: Sentences retrieved for the entity pair "*Ben Stiller*" and "*Mystery Men*", labeled in YAGO with relation ActedIn and respective degree of explicitness: explicit, implicit or not at all.

We use this approach to generate 5 different ground truth datasets for the two evaluation tasks. For the clustering task, we generate 3 datasets of approximately 200.000 sentences, each with a different number of distinct relations. For the information extraction task, we generate two small gold standard corpora that are manually checked for correctness, with all falsely labeled sentences filtered out: **GOLD**, a corpus of 300 sentences that explicitly express the labeled relation, and **SILVER**, a corpus of 400 sentences that either explicitly or implicitly express the labeled relation. Refer to Table 3 for a list of all datasets.

dataset	# sentences	# relations	# entity pairs	manually cleansed
R10	200.000	10	12.000	false
R20	200.000	20	9.000	false
R30	200.000	30	6.000	false
GOLD	300	20	300	true
SILVER	400	20	400	true

Table 3: Datasets created using YAGO and distant supervision. The three large datasets differ in number of distinct relations and contained entity pairs. **GOLD** and **SILVER** are smaller, manually cleaned datasets.

4.1.2 Measures

We use **BCubed** for extrinsic clustering evaluation (Amigó et al., 2009), an effective measure extendable to overlapping clustering, which satisfies the following essential criteria for measuring cluster quality [7]:

- *Cluster homogeneity*, which rewards clusterings with pure clusters.
- *Cluster completeness*, which promotes "same label, same cluster" policy.
- *Rag bag*, which rewards introducing a garbage cluster over polluting pure clusters.
- *Small cluster preservation*, which penalizes spreading data points of a rare label across various clusters.

General BCubed precision and recall are computed based on *Multiplicity*, a measure of the minimum intersection between two data points o_i and o_j regarding their labels and cluster assignments. In our case this intersection contains 1 element at most, since we performed

[7] Cf. (Han et al., 2011), chapter 10, for a more verbose elaboration on these quality criteria and BCubed in general.

non-overlapping clustering. Depending on whether precision or recall is computed, Multiplicity is normalized with the amount of shared cluster assignments or shared labels respectively:

$$Multiplicity_{precision}(o_i, o_j) = \frac{min(|C(o_i) \cap C(o_j)|, |L(o_i) \cap L(o_j)|)}{|C(o_i) \cap C(o_j)|} \tag{4}$$

$$Multiplicity_{recall}(o_i, o_j) = \frac{min(|C(o_i) \cap C(o_j)|, |L(o_i) \cap L(o_j)|)}{|L(o_i) \cap L(o_j)|} \tag{5}$$

Here $C(o_i)$ denotes the set of cluster assignments of a data point o_i given a clustering and $L(o_i)$ the set of labels for a given data point o_i according to ground truth. Precision and recall are then calculated by averaging Multiplicity over all data points [8]:

$$Precision_{BCubed} = \frac{\sum_{i=1}^{n} \frac{\sum_{o_j : C(o_i) \cap C(o_j) \neq \emptyset} Multiplicity_{precision}(o_i, o_j)}{\| \left\{ o_j | C(o_i) \cap C(o_j) \neq \emptyset \right\} \|}}{n} \tag{6}$$

$$Recall_{BCubed} = \frac{\sum_{i=1}^{n} \frac{\sum_{o_j : L(o_i) \cap L(o_j) \neq \emptyset} Multiplicity_{recall}(o_i, o_j)}{\| \left\{ o_j | L(o_i) \cap L(o_j) \neq \emptyset \right\} \|}}{n} \tag{7}$$

In a final step $Precision_{BCubed}$ and $Recall_{BCubed}$ are combined to give the F_1-score.

4.2 Clustering Task

We evaluate our method's ability to identify relations by comparing it on ground truth datasets of different composition with several baselines. We use BCubed F-measure to judge overall clustering performance.

4.2.1 Baselines

We compare our feature generation method (referred to as **PROP**) to the three baseline approaches using shallow analysis that were introduced in Section 2: The first is based on (Turney, 2011) and (Rosenfeld and Feldman, 2007) and uses a lexical feature generation technique with arbitrary word skips. We refer to this approach as **TUR**. A second baseline is modeled after (Bollegala et al., 2010), uses shallow lexico-syntactic patterns including pre- and postfix spans, and is referred to as **BOL**. The third baseline, after (Wang et al., 2011), uses lexical patterns without word skips and incorporates named entity class information. Patterns containing the verbs *to say* or *to tell* are filtered. This method is referred to as **WAN**.

[8] Note that self-relation is not excluded. And that Multiplicity is defined only when the two data points share at least 1 cluster assignment or label respectively.

4.2.2 Results

The results of the comparative evaluation are visualized in Figure 2. It clearly shows the impact of using an informed feature generation method. On the R30 dataset, we note overall increases in F-measure of 65% over the next best approach. Overall F-measure is highest around a k of 30 at **0.445**. The next best approach is WAN at $k = 12$ with **0.288**. This shows that our feature generation algorithm is capable of finding patterns for many expressions that shallow feature generation methods miss. Also, as F-measure peaks around $k = 30$, the results indicate that the clustering mechanism can effectively model the relations contained in the corpus.

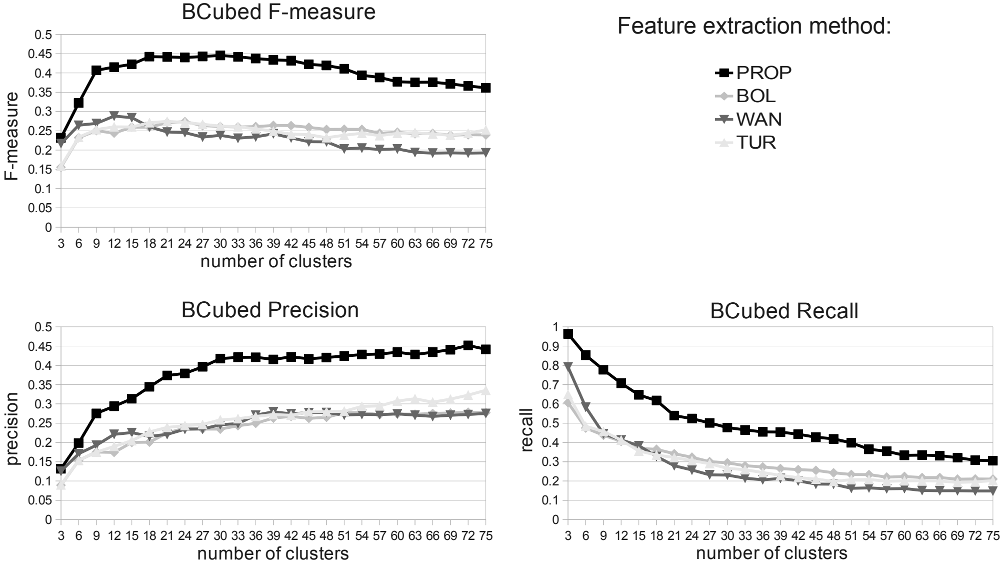

Figure 2: Clustering quality in terms of precision, recall and F_1-measure on the R30 dataset. The proposed feature generation approach (black line, square data points) outperforms all baseline approaches.

In order to examine this observation more closely, we use our approach on the R20 and R10 datasets, which contain 20 and 10 distinct relations respectively in the same amount of sentences as R30. The results are shown in Figure 3. Compared to results on the R30 dataset, we measure strong increases in F-measure on the R10 dataset which may be due to a much larger amount of examples per relation. However, we note that the results on the R20 and R30 are roughly similar, even though they are of different relational composition.

To gain more insight into these results, we inspect the data manually by randomly selecting clusters at different k. We make a number of observations. Firstly, when increasing k, the resulting clusters represent finer granularities of relations. The YAGO relation ChildOf, for example, is split into two clusters at higher k one representing the relation SonOf, the other DaugherOf. Similarly, the relation Created is split into multiple clusters, representing CreatedFilm, CreatedMusic and CreatedNovel respectively. The YAGO relation LocatedIn is split at various k into clusters of finer granularities, first into CityLocatedIn and VillageLocatedIn, then at an even higher k also into RiverLocatedIn. The YAGO relation IsAffiliatedTo is split at higher k into AffiliatedToSportsTeam and AffiliatedToPoliticalParty. These observations

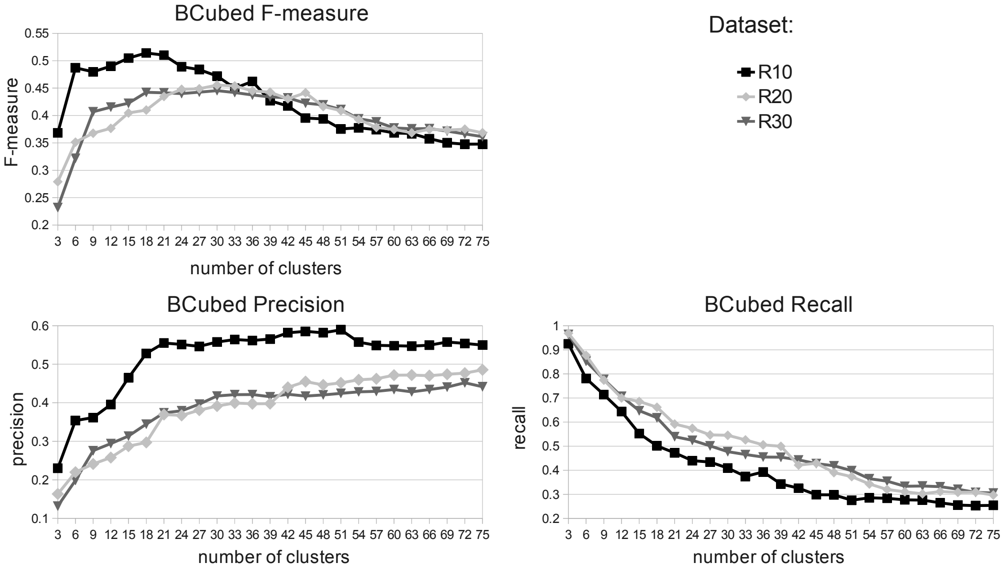

Figure 3: Clustering quality for the proposed approach on different datasets. Performance is best on dataset R10 which consists of relations typically expressed explicitly. R20 and R30 datasets each contain various more implicit or difficult to detect relations, causing lower F-measure.

indicate that the approach can be directed through parametrization to discover relations of varying granularity. However, we also note that some clusters split for difficult to interpret reasons, such as MarriedTo which splits into two clusters at higher k.

Generally, we observe that some relations are easier to identify by URE than others; relations like HasWonPrize or MarriedTo are often explicitly expressed and therefore easier to cluster. Other relations, like LivesIn are often, if at all, very implicitly expressed causing difficulties to the algorithm. Other relations, such as DealsWith, which signifies trade relations between two countries, are almost impossible to find as very few sentences express this relation. Again other relations, such as KnownFor are semantically very broad as there are any number of accomplishments (expressed in any number of ways) a person may be known for. This causes problems for a clustering approach. However, our approach is capable of finding a subset of all KnownFor relations in a cluster that resembles the InventorInvented relation. This indicates that not only the amount of distinct relations in a corpus is important, but also how explicitly they are expressed and whether one relation dominates a given entity pair. We also find that the most highly ranked patterns usually characterize clusters very well. Example clustering results are shown in Table 4. Here, we find examples of explicit patterns, entailment and implicit patterns. We examine the patterns more closely in the information extraction task.

4.3 Information Extraction Task

We evaluate the ranked patterns on the gold and silver datasets using the generated patterns as classifiers. If the classifier finds a known pattern in a sentence it extracts and labels a relation, but only if the distinctiveness of the pattern is above the classifier's *threshold* setting. We compute a precision-recall curve for our proposed approach over a range of threshold values

ID	example entity pairs	example patterns	YAGO label
(1)	- Media General; WJAR - CBS Radio; WPHT - News Corporation; Fox	*1.* **Y** owned by **X** *3.* **X** operate **Y** *5.* **X** acquire **Y** *6.* **X** parent company of **Y** *32.* **X** gain for buying **Y**	[OWNS]
(2)	- Ronny Yu; Fearless - John Madden; Proof - Dana Brown; Highwater	*1.* **Y** film directed by **X** *2.* **Y** directed by **X** *7.* **X** 's film **Y** *14.* find trailer info for **Y** by **X**	[DIRECTED]
(3)	- Alan Turing; Turing test - Carlos Chagas; Chagas disease - Hans Geiger; Geiger counter	*1.* **Y** invented by **X** *2.* **X** creator of **Y** *3.* **Y** discovered by **X** *7.* **Y** named after **X** *19.* **X** inventor known for invention of **Y**	[KNOWNFOR]

Table 4: Example of clustering output. Each cluster contains a set of entity pairs and is defined via a ranked list of patterns. The rank is given in italics before each pattern. The YAGO relation labels are not part of the clustering output and added for evaluation purposes only.

from 0 to 1, see Figure 4. The results show that the threshold can be used to control the tradeoff between precision and recall. If the threshold is set high, the extractor only uses patterns with high distinctiveness and finds relations at high precision and lower recall. At lower threshold settings, recall gradually increases while precision decreases. This indicates that the proposed method computes a valid ranking of patterns. The ability to use such a threshold setting to influence the precision-recall tradeoff is a valuable feature for information extraction.

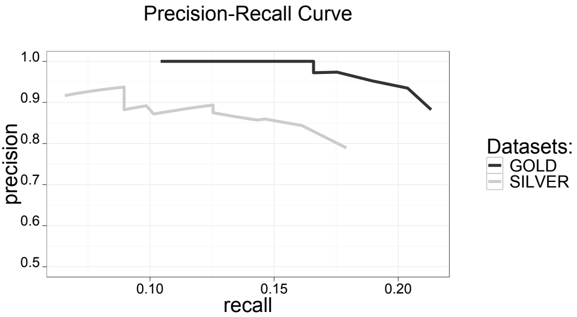

Figure 4: Precision-recall curves on the GOLD and SILVER gold standards for a range of threshold values. By lowering the confidence threshold, we trade high precision for an increase in recall.

We compare the approach against a reimplementation of (Lin and Pantel, 2001) in which patterns are directly clustered according to the distribution of entity pairs as introduced in (Sun and Grishman, 2010). This baseline (referred to as **PAN**) produces a hard clustering of patterns, without a distinctiveness value that can be used as threshold. We therefore compare the proposed approach against this baseline at two threshold levels: A threshold of 0 (PROP-0) and 1 (PROP-1). For completeness, we also compare the proposed approach using **TUR** and **WAN** instead of our proposed feature generation method. This is denoted as **PROP-TUR** and **PROP-WAN** respectively, again at threshold settings of 0 and 1.

	GOLD			SILVER		
approach	**precision**	**recall**	**F_1-measure**	**precision**	**recall**	**F_1-measure**
PROP-0	0.88	**0.21**	**0.34**	0.79	**0.18**	**0.29**
PROP-1	**1**	0.1	0.18	**0.92**	0.07	0.13
PAN	0.58	0.15	0.24	0.51	0.1	0.17
PROP-WAN-0	0.6	0.03	0.06	0.56	0.05	0.09
PROP-WAN-1	0.19	0.1	0.13	0.21	0.11	0.14
PROP-TUR-0	0.28	0.14	0.19	0.26	0.14	0.18
PROP-TUR-1	0.31	0.18	0.23	0.29	0.17	0.21

Table 5: Results of the information extraction task on the GOLD and SILVER ground truths. The proposed approach outperforms the baseline in precision, recall and F_1-measure.

The results in Table 5 show that the proposed method outperforms the baseline. Shallow patterns, as used in **TUR** and **WAN**, are hardly usable for information extraction. Especially on the GOLD dataset, in which all relations are explicitly expressed, we note very high precision of the proposed approach. As expected, precision is lower on the SILVER dataset, which also includes implicit expressions of relations, but still higher than the baseline. Overall, the results show that the pattern-relation assignment of the proposed approach yields valuable results for the task of information extraction.

5 Conclusion

In this paper we have presented a method for Unsupervised Relation Extraction that discovers relations from unstructured text as well as finding a list of discriminative patterns for each discovered relation. We introduced a feature generation algorithm that utilizes dependency parse information and demonstrated that using an informed feature generation technique significantly improves overall clustering F-measure. We interpreted clustering results to produce a set of pattern-relation assignments weighted according to the distinctiveness of each assignment using the theory of Discriminative Category Matching. We demonstrated that the strength of an assignment indicates how reliably a pattern evokes a relation by using the patterns for information extraction at different confidence thresholds. We presented a thorough evaluation of both relation discovery and pattern ranking on 5 datasets of different composition. We believe our approach to be a promising step towards achieving the goals of URE.

Future work will focus on further evaluation on a range of different clustering algorithms in order to find an optimal approach. Specifically, we believe that using overlapping or fuzzy clustering algorithms may counterbalance problems of entity pair ambiguities. Furthermore, since using more samples positively effected clustering quality, we aim to scale up the method to large corpora and more broadly inspect the results at different levels of granularity.

Acknowledgements

We would like to thank our student Do Tuan Anh for his help with gathering training data. Alan Akbik and Priska Herger received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no ICT-2009-4-1 270137 'Scalable Preservation Environments' (SCAPE). Larysa Visengeriyeva, Holmer Hemsen and Alexander Löser received funding from the Federal Ministry of Economics and Technology (BMWi) under grant agreement "01MD11014A, 'MIA-Marktplatz für Informationen und Analysen' (MIA)".

References

Akbik, A. and Broß, J. (2009). Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns. In *1st. Workshop on Semantic Search at 18th. WWW Conference*.

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486.

Bollegala, D., Matsuo, Y., and Ishizuka, M. (2010). Relational duality: unsupervised extraction of semantic relations between entities on the web. In *WWW*, pages 151–160.

Bullinaria, J. and Levy, J. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.

De Marneffe, M., MacCartney, B., and Manning, C. (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam (2011). Open information extraction: The second generation. In *IJCAI*, pages 3–10.

Fung, G. P. C., Yu, J. X., and Lu, H. (2002). Discriminative category matching: Efficient text classification for huge document collections. In *ICDM*, pages 187–194.

Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.

Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., de Melo, G., and Weikum, G. (2011). Yago2: exploring and querying world knowledge in time, space, context, and many languages. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 229–232, New York, NY, USA. ACM.

Klein, D. and Manning, C. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Lin, D. and Pantel, P. (2001). Dirt: discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM.

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Rosenfeld, B. and Feldman, R. (2007). Clustering for unsupervised relation identification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 411–418. ACM.

Rush, A. and Petrov, S. (2012). Vine pruning for efficient multi-pass dependency parsing. In *NAACL '12),*.

Sun, A. and Grishman, R. (2010). Semi-supervised semantic pattern discovery with guidance from unsupervised pattern clusters. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1194–1202. Association for Computational Linguistics.

Turney, P. (2008). The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33(1):615–655.

Turney, P. (2011). Analogy perception applied to seven tests of word comprehension. *Journal of Experimental & Theoretical Artificial Intelligence*, 23(3):343–362.

Turney, P. D. (2006). Expressing implicit semantic relations without supervision. In *ACL*.

Wang, W., Besançon, R., Ferret, O., and Grau, B. (2011). Filtering and clustering relations for unsupervised information extraction in open domain. In *CIKM*, pages 1405–1414.

Zhang, Y. and Nivre, J. (2011). Transition-based dependency parsing with rich non-local features. In *ACL (Short Papers)*, pages 188–193.