

Effective Selectional Restrictions for Unsupervised Relation Extraction

Alan Akbik Larysa Visengeriyeva Johannes Kirschnick Alexander Löser

Technische Universität Berlin

Databases and Information Systems Group

Einsteinufer 17, 10587 Berlin, Germany

firstname.lastname@tu-berlin.de

Abstract

Unsupervised Relation Extraction (URE) methods automatically discover semantic relations in text corpora of unknown content and extract for each discovered relation a set of relation instances. Due to the sparsity of the feature space, URE is vulnerable to ambiguities and underspecification in patterns. In this paper, we propose to increase the discriminative power of patterns in URE using selectional restrictions (SR). We propose a method that utilizes a Web-derived soft clustering of n -grams to model selectional restrictions in the open domain. We comparatively evaluate our method against a baseline without SR, a setup in which standard 7-class Named Entity types are used as SR and a setup that models SR using a fine-grained entity type system. Our results indicate that modeling SR into patterns significantly improves the ability of URE to discover relations and enables the discovery of more fine-granular relations.

1 Introduction

In traditional approaches for Relation Extraction (RE), all target relations (such as BORNIN or HASWONPRIZE) need to be specified in advance. For each relation, an extractor is trained or manually created that finds relation instances in text (Jiang and Zhai, 2007). This process is expensive and usually involves manually labeling large amounts of training data, making it difficult to scale RE to large sets of relations. Worse, because target relations must be manually defined in advance, the usefulness of RE in corpora of unknown content is limited (Akbik et al., 2012). This limits their applicability to the open domain where a potentially unbounded number of relations may be expressed in text.

In contrast, Unsupervised Relation Extraction (URE) approaches do not require target relations to be pre-specified, and require no labeled training data (Rosenfeld and Feldman, 2007). Instead, they automatically *discover* prominent relations in a given text corpus and *extract* for each identified relation a list of relation instances. Current methods (Akbik et al., 2012; Yao et al., 2012) utilize a vector space model of semantics in which they group co-occurring pairs of entities (referred to as *entity pairs*) into clusters based on distributional evidence over observed patterns. Each cluster is interpreted as one discovered semantic relation and the entity pairs in each cluster as the instances of this relation.

Pattern ambiguities. However, a problem for such approaches is that patterns may be ambiguous in the sense that they point to more than one relation. The pattern “[X] GET [Y]”¹ for example may be observed for a person and a product (“*Jim got a new VW Beetle.*”), a person and a disease (“*Jim got H1N1.*”) or (colloquially) between a person and a difficult-to-understand topic (“*Jim finally got Game Theory.*”). Such ambiguous patterns can cause entity pairs that belong to different relations (such as <Jim, VW Beetle> and <Jim, H1N1>) to be falsely grouped into the same semantic relation. See Table 1 for a structured illustration of this example.

This is especially problematic because the number of observed patterns for each individual entity pair is usually disproportionately small compared to the space of all possible patterns. In such a sparse feature space, false evidence caused by ambiguities can potentially have a negative impact on

¹Patterns are denoted with the placeholders [X] for the *subject* entity, and [Y] for the *object* entity of the entity pair they are observed with. In this paper, we use lexico-syntactic patterns extracted from dependency trees. For readability reasons, we omit information on dependency links. [X] in this pattern is either a subject or an apposition to the word GET. Likewise, [Y] is its object.

Sentence	Entity pair	Pattern	Restricted Pattern
<i>Yesterday, Jim got a new VW Beetle.</i>	<Jim, VW Beetle>	[X] GET [Y]	[X:PERSON] GET [Y:PRODUCT]
<i>Jim got H1N1.</i>	<Jim, H1N1>	[X] GET [Y]	[X:PERSON] GET [Y:DISEASE]
<i>Jim finally got Game Theory.</i>	<Jim, Game Theory>	[X] GET [Y]	[X:PERSON] GET [Y:THEORY]

Table 1: Example of pattern generation from three sentences. Three entity pairs are observed that belong to different relations. For example <Jim, VW Beetle> may belong to a PERSONACQUIREPRODUCT relation and <Jim, H1N1> to a PERSONINFECTEDWITHDISEASE relation. Without selectional restrictions, however, the same pattern is observed for all entity pairs, giving false evidence that they share the same relation. With selectional restrictions, different patterns are correctly observed.

the overall relation extraction quality of a URE approach.

Selectional restrictions in patterns. One approach to this problem is to include information on selectional restrictions (SR) to the patterns to increase their discriminative power (Resnik, 1996). We could restrict the patterns to apply only to entities of certain semantic classes or types. So, instead of the pattern “[X] GET [Y]” for the above mentioned examples we might generate “[X:PERSON] GET [Y:PRODUCT]” for “*Jim got a new VW Beetle.*”, “[X:PERSON] GET [Y:DISEASE]” for “*Jim got H1N1.*” and so forth (see Table 1).

However, modeling selectional restrictions in URE is not trivial, as it is unclear what type system and what *granularity* of types are required. For example, the types of a standard NER tagger (PERSON, LOCATION, ORGANIZATION etc.) may be too coarse grained for the above example, not being able to distinguish between DISEASE and PRODUCT.

While more fine-grained NER taggers have recently been researched (Ling and Weld, 2012), it is unclear whether they can be applied to the open domain. Here, we may encounter a potentially unrestricted set of entities of arbitrary types and granularity that varies from corpus to corpus. Also, each entity may have different types depending on how the type hierarchy is modeled; the string “*VW Beetle*” for instance may refer to a car, a product or a brand.

Contributions. In this paper, we address these challenges and study effective and viable methods for modeling selectional restrictions for URE in the open domain. We evaluate and discuss modeling SR using Named Entity types from the Stanford NER tagger (Finkel et al., 2005) as well as fine-grained Named Entity classes derived from the YAGO knowledge base (Hoffart et al., 2011). In addition, we propose a novel method that over-

comes shortcomings of existing methods by leveraging a Web-derived clustering of n -grams to model restrictions in an unsupervised fashion. We evaluate all setups against an informed baseline (based on previous work by (Akbik et al., 2012; Rosenfeld and Feldman, 2007)) in which patterns are not restricted.

We observe in all experiments that selectional restrictions significantly improve URE. The best performing setups use fine-grained Named Entity classes and our proposed open domain method, yielding f -measure improvements of 28% and 15% respectively over the baseline. We inspect the clustering results and find that the choice of SR influences the granularity of discovered relations. Based on our findings, we identify limitations of SR and outline challenges for URE.

2 Previous Work

We review previous work in URE with regards to selectional restrictions, and introduce the phrasal clustering dataset we use in our proposed method.

URE. There are a number of canonical works that relate to URE; (Lin and Pantel, 2001) first used distributional evidence to measure the similarity of patterns to find paraphrases of patterns. (Turney, 2006) instead computed the similarity of pairs of nouns using patterns as features. Their goal was finding analogies in text. (Rosenfeld and Feldman, 2007) then used a clustering method on a similar vector space model to group pairs of entities into clusters that represent semantic relations.

More recent work has addressed the problem of ambiguous patterns in URE in different ways. Notably, (Akbik et al., 2012) have evaluated pattern generation methods using lexical, shallow and deep syntactic features. They found that the use of deep syntactic features reduces pattern ambiguity and dramatically increases overall relation extraction f -measure by 65%. However, they do not model selectional restrictions in their pattern gen-

eration step.

Selectional Restrictions. Other recent work has incorporated information from NER taggers into their feature set. (Mesquita et al., 2010) use a standard 4-class NER tagger, but do not individually evaluate its impact. (Yao et al., 2012) use a very rich feature set, including fine-grained Named Entity types and document topics, to first disambiguate each pattern individually and in a second step perform URE using disambiguated patterns. This approach is problematic for many corpora because it requires a massive redundancy of pattern observations for disambiguation. In their experiments, they handled only patterns that are seen more than 200 times in their corpus. For comparison, in the large data set that we use in this paper, only 9 out of over 36,000 patterns are observed more than 200 times.

Phrasal Clustering. Contrary to previous work we do not propose using a manually established type system for selectional restrictions. Rather, we use a clustering of more than 10 million distinct one-to-five-word-grams from the Google n -gram data set (Lin and Wu, 2009) computed by (Lin et al., 2010). Previous work has leveraged the latent semantic information given by phrasal cluster memberships of n -grams to solve tasks other than URE. For example, (Zhou et al., 2011) increase the performance of deep syntactic parsers with regard to long-range dependencies, and (Täckström et al., 2012) transfer linguistic structure using cross-lingual word clusters.

In this work, we interpret each phrasal cluster as an entity type and all n -grams assigned to a cluster as belonging to this type. We incorporate this into the pattern generation step of our URE method and use this information to model selectional restrictions. Thus, the type system is not manually specified, but rather induced without supervision from a large Web corpus, making it a natural fit for the open domain and URE.

3 Pattern Generation

Pattern generation is the phase in URE in which patterns are generated for each co-occurring entity pair in the observed corpus. Current techniques go through each sentence in the corpus individually and generate <entity pair, pattern, count> tuples. In the following, we present the architecture of our URE system (Section 3.1) and illustrate how we integrate different options for modeling SR into

the pattern generation process. We present options that use types from an NER tagger (Section 3.2), fine-grained entity types from the YAGO knowledge base (Section 3.3), as well as the proposed phrasal clustering method (Section 3.4).

3.1 Baseline System

In our system, we use a pattern generation method that makes use of dependency parses. We implement the algorithm described in (Akbik et al., 2012). Here, patterns are generated as a sequence of typed dependencies and lemmas of tokens on the shortest path between two entities in a parse. In addition to the tokens on the shortest path (referred to as *core tokens*), additional tokens are collected from their vicinity in the dependency tree. The position of the entities are denoted by the placeholders “[X]” and “[Y]”. We further prune patterns using linguistically-informed filters, e.g. removing patterns that consist only of direct dependencies between subject and object. We give an example of pattern generation applied to a sentence in Figure 1.

Similarity of entity pairs. Using this technique, we generate a list of pattern-entity pair observation tuples, which we use to construct a pair pattern frequency matrix. Each row vector represents one distinct entity pair e_i and each column one distinct pattern p_j . The value of the matrix cell c_{ij} is the number of times that e_i occurs in the pattern p_j . This representation allows us to compute the similarity of two entity pairs by computing the cosine distance between their corresponding rows in the pair pattern matrix (Bullinaria and Levy, 2007). We compute the pairwise similarity for all entity pairs to generate a dissimilarity matrix and execute a clustering method on this matrix.

Clustering. In line with most previous work in URE (Rosenfeld and Feldman, 2007; Wang et al., 2011), we use a Hierarchical Agglomerative Clustering (HAC) approach with the *average linkage* scheme (Han et al., 2011). This approach iteratively merges the two closest entity pairs to compute a dendrogram of cluster merges.

The dendrogram is cut at a point given by the *cutting threshold* parameter, yielding a set of clusters. This parameter is usually estimated or determined through experimentation. A common method is to execute an exhaustive search over a subset of the parameter space (referred to as

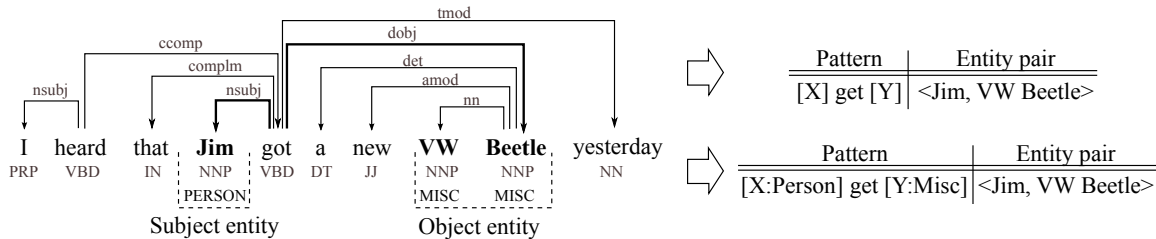


Figure 1: Illustration of the pattern generation process for one example sentence with the entities “Jim” and “VW Beetle”. Part-of-speech and Named Entity class tags are given below the tokens in the sentence. The shortest path between the two entities is highlighted bold in the dependency tree. The word “got” lies along this path, which is lemmatized to produce the pattern [X] GET [Y]. As an option, the Named Entity classes of the entities are included as selectional restrictions into the pattern, yielding the pattern [X:PERSON] GET [Y:MISC].

grid search), guided by cross-validation on a training set (Bergstra and Bengio, 2012). Through such experimentation, we determine that the cutting threshold must be set at a high value (for example around 0.999) to produce good clustering results².

Clustering result. The clustering produces a set of clusters, which each consist of a set of entity pairs. Each resulting cluster is interpreted as one discovered relation, and all entity pairs in the cluster as the instances of this relation. The clustering result is then passed to an evaluation step discussed in detail in Section 4.

3.2 Named Entity Type Restrictions

We first extend the system with an option to use standard Named Entity types as selectional restrictions, in a similar fashion as a previous URE system (Mesquita et al., 2010). We incorporate the Stanford NER 7-class tagger into the sentence parsing pipeline and determine the type of each entity. These types are used to restrict the generic placeholders [X] and [Y] in generated patterns with the types of the subject and object entities.

For the example sentence illustrated in Figure 1, the tagger determines the class PERSON for “Jim”, and MISC for “VW Beetle”. The latter class is used for all entities that cannot be assigned any of the named classes. We therefore generate the pattern “[X:PERSON] GET [Y:MISC]” in this example. Because we model entity type restrictions directly into the patterns, we increase the space of all possible patterns and make individual patterns

²We determine different values for other linkage schemes. For example, when using the *single-link* linkage scheme in HAC, we find a good estimation for the cutting threshold to be 0.9.

more discriminative.

However, as shown in the example in Section 1, the Named Entity classes given by a 7-class tagger are coarse grained and may not include the types necessary to disambiguate all patterns. Also, there is a risk that Named Entity taggers may determine the wrong type for an entity³. This could lead to false evidence that negatively impacts URE.

3.3 Fine-grained Entity Type Restrictions

Because classes from a standard 7-class NER tagger may be too coarse grained for URE, we next extend the system with the option of modeling fine-grained Named Entity classes. We choose an approach that requires entities to be disambiguated and linked to Wikipedia URIs. The YAGO knowledge base then enables us to retrieve fine-grained entity classes for disambiguated entities, such as their Wikipedia categories (of which we use only the head nouns as restrictions). Because many YAGO entities belong to more than one class, this method returns a set of classes for each entity. For example, the Wikipedia categories for “VW Beetle” are “TAXICAB VEHICLES”, “AUSTRIAN INVENTIONS”, “INDUSTRIAL DESIGNS”, “SUBCOMPACT CARS” and others.

For each entity pair, we retrieve two sets of entity classes (one for the subject and one for the object). We determine the Cartesian product over these two sets and create one distinct pattern with selectional restrictions for each combination. For the example sentence, this means that we generate a list of patterns, including “[X:PERSON] GET [Y:CAR]”, “[X:PERSON] GET [Y:VEHICLE]” and “[X:PERSON] GET [Y:INVENTION]”, each of

³(Finkel et al., 2005) report an overall *f*-measure of 87% on the CoNLL 2003 Named Entity Recognition dataset.

1) Lookup phrasal clusters for subject and object entity

Pattern	Entity pair
[X] get [Y]	<Jim, VW Beetle>

<i>N</i> -gram lookup	Cluster	Weight	Other <i>n</i> -grams in cluster
VW Beetle	825	0.3	Chrysler Voyager, Toyota Highlander
VW Beetle	805	0.17	Computer Parts, Office Supplies
⋮	⋮	⋮	⋮
Jim	269	0.4	Becky, Doug, Eileen, Frances
⋮	⋮	⋮	⋮



2) Add cluster IDs as selectional restrictions to patterns.

Pattern	Entity pair	Weight
[X:269] get [Y:825]	<Jim, VW Beetle>	0.12
[X:269] get [Y:805]	<Jim, VW Beetle>	0.04
[X:283] get [Y:269]	<Jim, VW Beetle>	0.18
⋮	⋮	⋮

Figure 2: Illustration of the proposed pattern generation process that uses phrasal cluster memberships as selectional restrictions. In 1), phrasal clusters are retrieved for the subject and object of the entity pair. “VW Beetle” for example is in cluster 825, which contains many other car names. In 2) the Cartesian product over the phrasal clusters for subject and object is built and used as selectional restrictions for the pattern generated with the baseline method. This yields a set of patterns with different selectional restrictions.

which is used as a feature. While this method increases the overall number of observed patterns by about one order of magnitude, individual patterns are much more discriminative than without selectional restrictions.

Limitations. Two things must be noted regarding this method of determining fine-grained Named Entity classes. Firstly, it does not necessarily produce patterns at the desired granularity. In Section 1 we discussed the pattern “[X:PERSON] GET [Y:PRODUCT]” to be most appropriate, which is not generated by this method. More importantly though, the method is limited to entities that can be disambiguated to the appropriate Wikipedia page. While this is possible on the dataset we use for the evaluation, it is much more difficult to determine fine-grained Named Entity classes in the open domain with this method. We therefore implement this option mainly for evaluation purposes, in order to determine URE capabilities given a fine-grained, high quality type system for selectional restrictions.

3.4 Phrasal Clusters as Restrictions

To address the limitations of the methods described in 3.3, we propose a method for modeling SR that does not require an existing type system or the disambiguation of entities.

We extend the system with the option of using selectional restrictions derived from a phrasal

clustering computed by (Lin and Wu, 2009) over a dataset of more than 10 million distinct one-to-five-word-grams from the Google *n*-gram data set (Lin and Wu, 2009). In this dataset, each *n*-gram is assigned to ten different phrasal clusters with different association values, also referred to as *weights*. Weights are between 0 and 1, with a higher value indicating a stronger assignment confidence. Because the clustering is based on lexical context, *n*-grams in a cluster often share semantic properties. For example, the dataset contains clusters of entities like cities, cars, movies, etc (Lin and Wu, 2009).

During pattern generation, we look up the phrasal cluster IDs for the lexical representation of an entity and use this ID as selectional restriction. For example, the string “VW Beetle” belongs to phrasal cluster number 825 with weight 0.3. Semantically similar strings, such as “Chrysler Voyager” and “Toyota Highlander” are also part of this cluster. We can use this information to restrict the subject of the pattern only to strings that belong to cluster 825. Another phrasal cluster for “VW Beetle” is cluster 805 (with a lower weight of 0.17), which consists of more general product terms such as “Computer Parts” and “Office Supplies”. “Jim” is found in cluster 269, which contains many person first names. See Figure 2 for an illustration of this example.

We build the Cartesian product over the two

sets of phrasal clusters retrieved for the subject and object of an entity pair. Because each entity (e.g. its lexical representation) has 10 soft cluster memberships, the Cartesian product of phrasal clusters for both entities of an entity pair yields a total of 100 distinct weighted phrasal cluster ID combinations, hereafter referred to as *restriction pairs*. The weight of each restriction pair is computed by building the product of the confidence weights of the respective entity-phrasal cluster assignments. Each restriction pair is encoded into its pattern by adding to the entity placeholders “[X]” and “[Y]” a qualifier indicating the phrasal cluster ID. For each observation and restriction pair, a distinct pattern is generated.

This option increases the overall number of distinct patterns by two orders of magnitude. Patterns are also less humanly readable than their counterparts that use coarse- or fine-grained Named Entity types. We use this feature space to evaluate the assumption that we can leverage distributional evidence over a large Web corpus to model selectional restrictions in URE without an existing type system.

4 Evaluation

In this section, we perform experiments to measure the impact of different options of modeling selectional restrictions in patterns for URE. We also qualitatively inspect clusters and patterns.

4.1 Experimental Setup

Our experiments are performed on a silver standard dataset of 200,000 sentences crawled from the Web and labeled using distant supervision (Mintz et al., 2009). The sentences contain 4500 distinct entity pairs that are part of the YAGO knowledge base⁴. This allows us to compare the results of URE against the YAGO knowledge base. We compute BCubed (Amigó et al., 2009) precision, recall and f -measure values, which are commonly used to extrinsically evaluate clustering results. We perform this evaluation on the following setups:

BASELINE In this setup, we establish the URE quality of the baseline system (see Section 3.1) without modeling selectional restrictions. The baseline is based on the system described in (Akbik et al., 2012).

⁴In (Akbik et al., 2012) we illustrate and evaluate the labeling procedure in detail.

NER-7CLASS This scenario simulates previous work by (Mesquita et al., 2010). We evaluate the impact of using a standard NER tagger to model selectional restrictions (see Section 3.2).

NER-YAGO In this setup, we evaluate the use of a high quality, fine-grained type system to model selectional restrictions. We retrieve fine-grained entity classes from Wikipedia categories as described in Section 3.3.

PROPOSED-OPEN-1 This setup is a modification of the proposed method that makes use of phrasal clusters to model selectional restrictions in the open domain. Here, we only use the cluster with the top weight (instead of all 10) as restriction for an entity.

PROPOSED-OPEN-5 Like PROPOSED-OPEN-1 this is a modification of the proposed method. Here, the top 5 clusters for each string are used as restrictions. We use this setup to assess the impact of using only the most likely portion of the full phrasal clusters data set.

PROPOSED-OPEN-FULL The proposed method making use of the full phrasal clusters data set.

In addition to varying the pattern generation method we also experiment with different cutting thresholds in the Hierarchical Agglomerative Clustering method. We use two cutting threshold parameters that were determined through experimentation (see Section 3.1), namely 0.9995 and 0.9999 (referred to as $C_{0.9995}$ and $C_{0.9999}$ respectively). We also perform a grid search over the parameter space to determine the best cutting threshold for each setup, which we refer to as C_{best} .

4.2 Quantitative Evaluation

Table 2 shows the results of the quantitative evaluation. At all cutting thresholds, we observe improvements in overall f -measure with all setups (except PROPOSED-OPEN-1) over the baseline. These results indicate the value of including selectional restrictions in the pattern generation step of a URE method. When comparing the different methods, we note that NER-YAGO and PROPOSED-OPEN-FULL perform best, outperforming the baseline at peak setting by 15% and 28% respectively. PROPOSED-OPEN-1 performs much worse than the baseline, especially

	$C_{0.9995}$			$C_{0.9999}$			C_{best}		
	P	R	F1	P	R	F1	P	R	F1
BASELINE	0.34	0.59	0.43	0.21	0.74	0.33	0.46	0.45	0.46
NER-7CLASS	0.39	0.55	0.46	0.52	0.45	0.48	0.51	0.47	0.49
NER-YAGO	0.74	0.39	0.52	0.65	0.50	0.57	0.65	0.53	0.59
PROPOSED-OPEN-1	0.95	0.02	0.04	0.95	0.02	0.04	0.95	0.02	0.04
PROPOSED-OPEN-5	0.70	0.31	0.43	0.59	0.45	0.51	0.57	0.49	0.53
PROPOSED-OPEN-FULL	0.58	0.45	0.51	0.49	0.54	0.51	0.57	0.49	0.53

Table 2: Overview of the results of the comparative evaluation. At all cutting threshold settings, setups NER-YAGO and PROPOSED-OPEN-FULL achieve significantly higher f -measure scores than the baseline. We find that at peak performance, the PROPOSED-OPEN-5 setup reaches a similar quality as PROPOSED-OPEN-FULL.

BASELINE		
ID	Example patterns	Example entity pairs
1	[Y] OWNED BY [X], [Y] PART OF [X],	[X] BUY [Y], [Y] ACQUIRED BY [X] <SNCF, Systra> <Eskom, Arnet Power Station>
2	[X] WIN [Y], [Y] WINNING [X],	[X] RECEIVE [Y], [X] NOMINATED FOR [Y] <Cher, Emmy Award> <Chile, Chilean War for Independence>
3	[X] 'S SON [Y], [X] FATHER OF [Y]	[Y] BORN TO [X], [X] DAUGHTER OF [Y] <Zeus, Heracles> <Carus, Carinus>
4	[X] CREATE [Y], [Y] INVENTED BY [X],	[Y] BY [X], [Y] DEVELOPED BY [X] <Philipps, Compact Disc> <Kent Beck, Extreme Programming>
NER-YAGO		
ID	Example patterns	Example entity pairs
5	[X:FORMATIONS] FIGHT IN [X:WARS], [X:ORGANIZATIONS] WIN [Y:WARS], [Y:CONFLICTS] BETWEEN [X:ORGANIZATIONS]	<Red Army, Russian Civil War> <Rebel Alliance, Galactic Civil War>
6	[Y:PEOPLE] STUDENT OF [X:PHILOSOPHERS], [Y:PEOPLE] INFLUENCED BY [X:PHILOSOPHERS], [X:PHILOSOPHERS] TEACHER OF [Y:PEOPLE]	<Aristotle, Maimonides> <Ayn Rand, Ron Paul>
7	[Y:ALBUMS] BY [X:SINGERS], [Y:ALBUMS] ALBUM BY [X:MUSICIANS], [Y:ALBUMS] PERFORMED BY [X:MUSICIANS]	<Lou Reed, Coney Island Baby> <Bryan Adams, Reckless>
8	[Y:VENUES] HOME OF [X:TEAMS], [X:CLUBS] PLAY AT [Y:STADIUMS], [Y:CLUBS] AT [X:LOCATIONS]	<Milwaukee Brewers, Miller Park> <New York Yankees, Yankee Stadium>
PROPOSED-OPEN		
ID	Example patterns	Example entity pairs
9	[X:204] IN [Y:809], [X:809] IN [Y:764],	[X:204] IN [Y:764], [X:203] IN [Y:809] <Bob Gale, Back to the Future> <Cher, Zookeeper (film)>
10	[X:18] [Y:452] CANDIDATE, [X:793] [Y:284] CANDIDATE,	[X:233] [Y:441] POLITICIAN, [X:259] [Y:441] POLITICIAN <Bob Allen, Republican Party> <Fob James, Democratic Party>

Table 3: 10 sample clusters found with setups BASELINE, NER-YAGO and PROPOSED-OPEN. Each cluster is characterized by the top patterns in its centroid and represents one discovered relation. The entity pairs that make up the cluster are instances of discovered relations. Cluster 3, for example, represents the CHILDOF relation which holds between two persons.

with regards to recall. This is because this method produces highly overspecified patterns that do not allow for efficient grouping of entity pairs.

We also note that the cutting threshold setting has a significant impact on recall, precision and f -measure. At the PROPOSED-OPEN-5 setup, for example, minor variations in the parameter (from $C_{0.9995}$ to $C_{0.9999}$) cause an absolute f -measure difference of 0.8 points. This observation strongly indicates the importance of finding methods to ef-

fectively parameterize URE.

4.3 Qualitative Evaluation

We manually inspect a sample of the discovered relations and patterns to gain insight into how the different setups affect the relation discovery capabilities of our URE method. We illustrate our observations with a number of clusters shown in Table 3. We give examples of clusters for three setups: BASELINE, NER-YAGO and PROPOSED.

For each cluster, which represents one discovered relation, we list a small set of representative patterns and entity pairs.

Cluster 1, for example, is a cluster that represents company acquisitions, as is indicated by top patterns such as “[Y] ACQUIRED BY [X]” and “[Y] PART OF [X]”. Entity pairs in this cluster are relation instances. This means that <Eskom, Arnet Power Station> is an instance of the COMPANYACQUISITION relation. We find that this cluster corresponds most closely to the OWNS relation from the YAGO knowledge base.

Readability. Generally, we note that using named classes for selectional restrictions (NER-7CLASS and especially NER-YAGO) result in more human readable patterns than their counterparts in the baseline and proposed methods. Consider clusters 6 and 9. Cluster 6 is easy to evaluate, as the top patterns are human readable. It represents the INFLUENCEDBY relation that holds between a person and a philosopher. Cluster 9, on the other hand, is characterized by patterns that consist only of prepositions and phrasal cluster IDs. We must consult the entity pairs to determine that this cluster represents the ACTEDIN relation that holds between an actor and a film.

Granularity. In many cases, we find that selectional restrictions lead to the discovery of more fine-grained relations. An example of this is cluster 7, which denotes a relationship between a singer and the music album she created. All 46 instances in this cluster belong to the more general CREATED relation from YAGO that holds between a person and something she created (such as films, novels, albums etc.). This observation has implications for the use of selectional restrictions in URE, namely that the granularity of discovered relations can be influenced by the choice of type system. This also points to difficulties for the method of evaluating URE against an existing knowledge base as discovered relations might differ in granularity from the KB schema. Both these observations merit further investigation in future work.

Ambiguities. We look into errors made by the URE method and find that many errors are due to pattern ambiguities. Cluster 2, for example, mostly corresponds to the YAGO relation HASWONPRIZE. However, the patterns “[X] WIN [Y]” and “[Y] WINNING [X]” that hold between correct instances such as <Cher, Emmy Award>

also hold between false positives such as <Chile, Chilean War for Independence>. In more discriminative setups, this error is not made. For example, cluster 5 contains the more differentiated pattern “[X:ORGANIZATIONS] WIN [Y:WARS]”.

5 Conclusion and Outlook

In this paper, we addressed the problem of pattern ambiguities in URE by evaluating different methods of modeling selectional restrictions. We find that SR generally have a positive impact on relation discovery capabilities of our URE method. Significantly, we find a fine-grained type system to be the best setting, especially if URE is applied to a closed domain where most types of interest can be detected. For the open domain, we have presented a method that makes use of a Web-derived phrasal clustering of n -grams. We find our proposed method to be effective in reducing pattern ambiguities, with the advantage of being independent of a manually determined type system. Based on our results, we believe that correctly restricted deep syntactic patterns are the best features for URE.

In a qualitative evaluation of clustering results, we have determined two main issues that merit being addressed in future work in URE. First, automatic evaluation of URE remains problematic, as relations might be discovered that differ in granularity or semantics from the knowledge base that is evaluated against. Current evaluation methods penalize such divergence, even though discovered relations might still be correct. Second, we found that the parameterization of the clustering approach used in URE greatly influences the result quality and granularity. We find that even minor variations on the cutting threshold parameter for Hierarchical Agglomerative Clustering greatly impact overall f -measure.

Future work will focus on closely investigating clustering techniques and methods for effective parametrization. In addition, we intend to investigate Active Learning (Sun and Grishman, 2012) as a method to include minimal amounts of human feedback to guide the relation discovery process and improve overall URE results.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. Alan Akbik received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no ICT-2009-4-1 270137 ‘Scal-

able Preservation Environments' (SCAPE). Larysa Visengeriyeva, Johannes Kirschnick 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

- A. Akbik, L. Visengeriyeva, P. Herger, H. Hemsén, and A. Löser. 2012. Unsupervised discovery of relations and discriminative extraction patterns. In *Proceedings of the 24th International Conference on Computational Linguistics*.
- Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486, August.
- James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13:281–305.
- J.A. Bullinaria and J.P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.
- Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.
- Jiawei Han, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.
- Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. 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.
- Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120.
- Dekang Lin and Patrick Pantel. 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.
- D. Lin and X. Wu. 2009. Phrase clustering for discriminative learning. 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*, ACL '09, pages 1030–1038, Stroudsburg, PA, USA. Association for Computational Linguistics.
- D. Lin, K. Church, H. Ji, S. Sekine, D. Yarowsky, S. Bergsma, K. Patil, E. Pitler, R. Lathbury, V. Rao, et al. 2010. New tools for web-scale n-grams. In *Proceedings of LREC*.
- Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*.
- F. Mesquita, Y. Merhav, and D. Barbosa. 2010. Extracting information networks from the blogosphere: State-of-the-art and challenges. In *Fourth Int'l AAAI conference on weblogs and social media*.
- M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL/AFNLP*, pages 1003–1011.
- P. Resnik. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61(1):127–159.
- B. Rosenfeld and R. Feldman. 2007. Clustering for unsupervised relation identification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 411–418. ACM.
- Ang Sun and Ralph Grishman. 2012. Active learning for relation type extension with local and global data views. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1105–1112. ACM.
- O. Täckström, R. McDonald, and J. Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. *Proceedings of the North American Chapter of the Association for Computational Linguistics*.
- P.D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.
- W. Wang, R. Besançon, O. Ferret, and B. Grau. 2011. Filtering and clustering relations for unsupervised information extraction in open domain. In C. Macdonald, I. Ounis, and I. Ruthven, editors, *CIKM*, pages 1405–1414. ACM.
- Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012. Unsupervised relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 712–720. Association for Computational Linguistics.
- G. Zhou, J. Zhao, K. Liu, and L. Cai. 2011. Exploiting web-derived selectional preference to improve statistical dependency parsing. In *Proceedings of ACL*, pages 1556–1565.