N-ary Biographical Relation Extraction using Shortest Path Dependencies

Gitansh Khirbat **Jianzhong Qi** **Rui Zhang**
Department of Computing and Information Systems
The University of Melbourne
Australia
gkhirbat@student.unimelb.edu.au
{jianzhong.qi, rui.zhang}@unimelb.edu.au

Abstract

Modern question answering and summarizing systems have motivated the need for complex *n*-ary relation extraction systems where the number of related entities (n) can be more than two. Shortest path dependency kernels have been proven to be effective in extracting binary relations. In this work, we propose a method that employs shortest path dependency based rules to extract complex *n*-ary relations without decomposing a sentence into constituent binary relations. With an aim of extracting biographical entities and relations from manually annotated datasets of Australian researchers and department seminar mails, we train an information extraction system which first extracts entities using conditional random fields and then employs the shortest path dependency based rules along with semantic and syntactic features to extract *n*-ary affiliation relations using support vector machine. Cross validation of this method on the two datasets provides evidence that it outperforms the state-of-the-art *n*-ary relation extraction system by a margin of 8% F-score.

1 Introduction

Information extraction (IE) is the process of extracting factual information from unstructured and semi-structured data and storing it in a structured queryable format. Two important components of an IE system are entity extraction and relation extraction. These components are sequential and together form the backbone of a classic IE system. Entity extraction systems have achieved a high accuracy in identifying certain entities such as mention of people, places and organizations (Finkel et al., 2005). However, such *named entity recognition* (NER) systems are domain-dependent and do not scale up well to generalize across all entities.

Relation extraction systems utilize the identified entities to extract relations among them. Past two decades have witnessed a significant advancement in extracting binary domain-dependent relations (Kambhatla, 2004), (Zhao and Grishman, 2005) and (Bunescu and Mooney, 2005a). However, modern question answering and summarizing systems have triggered an interest in capturing detailed information in a structured and semantically coherent fashion, thus motivating the need for complex *n*-ary relation extraction systems (where the number of entities, $n \geq 2$). Some notable *n*-ary relation extraction systems are (McDonald et al., 2005) and (Li et al., 2015). McDonald et al. (2005) factorized complex *n*-ary relation into binary relations, representing them in a graph and tried to reconstruct the complex relation by making tuples from selected maximal cliques in the graph. While they obtained reasonable precision and recall using a maximum entropy binary classifier on a corpus of 447 selected abstracts from MEDLINE, they have not explored the constituency and dependency parse features which have been proven to be efficient in relation extraction. Li et al. (2015) make use of lexical semantics to train a model based on distant-supervision for *n*-ary relation extraction. However, the applicability of this method on other datasets is not clear.

We design an algorithm for extracting *n*-ary relations from biographical data which extracts entities using conditional random fields (CRF) and *n*-ary relations using support vector machine (SVM) from two manually annotated datasets which contain biography summaries of Australian researchers. Shortest path dependency kernel (Bunescu and Mooney, 2005a) has been proven to be the most efficient in extracting binary relations. In this work, we propose the use of shortest path

Gitansh Khirbat, Jianzhong Qi and Rui Zhang. 2016. N-ary Biographical Relation Extraction using Shortest Path Dependencies. In *Proceedings of Australasian Language Technology Association Workshop*, pages 74–83.

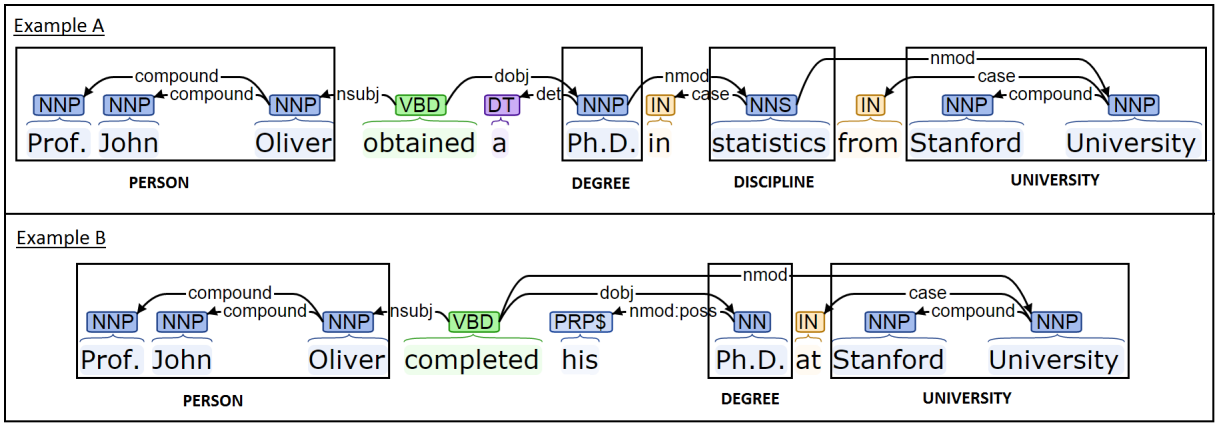

Figure 1: Example sentences with their dependency parses

dependency based rules to extract complex *n*-ary relations without decomposing the sentences into binary relations. These rules are based on the hypothesis which stipulates that the contribution of the sentence dependency graph to establish a relationship is almost exclusively concentrated in the shortest path connecting all the entities such that there exists a single path connecting any two entities at a given time. We present a thorough experimental evaluation and error analysis, making the following contributions:

- We propose a new approach to handle *n*-ary relation extraction using shortest path dependency-based rules.
- We conducted a thorough empirical error analysis of using CRF-based entity extractor coupled with SVM-based relation extractor.
- We present two manually annotated corpora containing biographical entities and relation annotations, which can be used for research or to augment existing knowledge bases.

The rest of the paper is organized as follows. Section 2 defines the problem. Section 3 reviews related studies. Section 4 discusses our methodology. Section 5 introduces the corpora. Section 6 presents the experiments. Section 7 presents an error analysis and Section 8 concludes this paper.

2 Preliminaries

2.1 *N*-ary Relation Extraction

We study the problem of *n*-ary relation extraction. A relation is defined in the form of a tuple $t =< e_1, e_2, \ldots, e_n >$ where e_i is an entity, which can be mention of a person, place, organization, etc. The most studied relations are binary relations, which involve two entities. If more than two entities exist in a relation, it becomes a complex relation which is called an *n*-ary relation. McDonald et al. (2005) define a complex relation as any *n*-ary relation among *n* entities which follows the schema $< t_1, \ldots, t_n >$ where t_i is an entity type. An instance of this complex relation is given by a list of entities $< e_1, e_2, \ldots, e_n >$ such that either type(e_i) = t_i, or $e_i = \perp$ indicating that the *i*th element of the tuple is missing. Here, type(e_i) is a function that returns the entity type of entity e_i.

For example, assume that the entity types are E={*person (PER)*, *degree (DEG)*, *discipline (DISC)*, *position (POS)*, *university (UNI)*} and we are interested to find a *n*-ary relation with schema <*PER, DEG, DISC, UNI*> that provides information of a *person* affiliated to a *university*, studying a *degree* in a *discipline*. In example A shown in Figure 1, the expected extracted tuple is <Prof. John Oliver, Ph.D., statistics, Stanford University>. In example B, the expected extracted tuple is <Prof. John Oliver, Ph.D., $\perp$, Stanford University>, since the *discipline* entity is not mentioned. Thus, *n*-ary relation extraction systems aim to identify all instances of a complete and partially complete relations of interest.

2.2 Problem Definition

Given a set of D documents containing biographical data, we classify words in a document $d_i \in D$ into entities $< e_1, e_2, \ldots, e_j >$ and *n*-ary relations given by dataset R, such that $r_k \in R$ is a tuple $t =< e_1, e_2, \ldots, e_n >$ where $n \geq 2$. In particular, we are interested in extracting affiliation relations such as the one mentioned in Section 2.1.

3 Related Work

Information extraction is a sequential confluence of two processes - entity extraction and relation extraction. Entity extraction refers to the task of NER wherein the task is to correctly classify an entity (like person, location, organization, etc.) out of a given sentence in a textual document. Past two decades have seen a massive body of work which aimed to improvise the entity extraction systems (Bikel et al., 1997), (Cunningham et al., 2002) and (Alfonseca and Manandhar, 2002). It is a well-explored research area which has reached maturity (Finkel et al., 2005). Most NER systems are domain dependent and require training with a new annotated corpus for a new task.

Relation extraction refers to the task of finding relations among the entities which were obtained during entity extraction. A huge body of work addresses the task of extracting binary relations wherein a relation exists between two entities only. Feature-based supervised learning methods like (Kambhatla, 2004) and (Zhao and Grishman, 2005) leverage the syntactic and semantic features. Exploration of a large feature space in polynomial computational time motivated the development of kernel based methods like tree kernels (Zelenko et al., 2003) and (Culotta and Sorensen, 2004), subsequence kernels (Bunescu and Mooney, 2005b) and dependency tree kernel (Bunescu and Mooney, 2005a). Open IE system (Banko et al., 2007) gives a sound method to generalize the relation extraction process, however the system does not give any insights to extract complex *n*-ary relations.

With advances in biomedical text mining and modern question answering systems, complex *n*-ary relation extraction is gaining attention wherein the task is to detect and extract relations existing between two or more entities in a given sentence. McDonald et al. (2005) attempt to solve this problem by factorizing complex relations into binary relations which are represented as a graph. This graph is then used to reconstruct the complex relations by constructing tuples from selected maximal cliques scored on the graph. Li et al. (2015) make use of lexical semantics to train a model based on distant-supervision for *n*-ary relation extraction. However, both these systems are computationally expensive and do not scale up efficiently.

Bunescu and Mooney (2005a) advocate the use of shortest path between the entities in a dependency parse to compute the cartesian product of dependencies clubbed with respective POS tags. This method has been proven to be the best among all kernel methods to extract binary relations. However, it is yet to be confirmed if it works for extracting complex *n*-ary relations.

4 Methodology

4.1 Shortest path dependency: binary to *n-ary* relations

We use dependency parsing (Manning et al., 2014) to help extract *n*-ary relations. Dependency parse provides information about word-word dependencies in the form of directed links. These dependencies capture the predicate-argument relations present in the sentence. The finite verb is taken to be the structural centre of the clause structure. All other syntactic units (words) are connected either directly (to the predicate) or indirectly (through a preposition or infinitive particle) to the verb using directed links, which are called dependencies. Each dependency consists of a *head* from where the directed link originates and a *dependent* where the link terminates. Dependencies can be classified into two categories - local and non-local dependencies. Local dependencies refer to the dependencies which occur within a sentence and can be represented by predicate-argument structure. Non-local dependencies refer to long-range dependencies involving two positions in a phrase structure whose correspondence can not be captured by invoking predicate-argument structure.

Bunescu and Mooney (2005a) successfully demonstrated the use of shortest path dependencies between two entities to extract *located (at)* relation. We extend this hypothesis to form shortest path dependency based rules for *n*-ary relation extraction. If a sentence has n entities $e_1, e_2, \ldots, e_n$ such that there exists a relation r among them, our hypothesis stipulates that dependency graph can be used to establish the relationship $r(e_1, e_2, \ldots, e_n)$ by leveraging the shortest path connecting all the entities such that there exists a single path connecting any two entities at a given time.

Entities are considered as one unit. In order to determine entity-level dependency of an entity e_i, the compound dependencies are discarded and the dependency between a word $\in e_i$ and the surrounding word $\notin e_i$ is considered. For any two consecutive entities in a sentence,

- If there exists a direct dependency between the two words belonging to two entities e_1 and e_2, it is represented as ($NER(e_1)$–*dependency name*–$NER(e_2)$). This happens mostly in the case of local dependencies. In Example A, it can be illustrated by (*Degree–nmod–Discipline*).

- If there exists a common word connecting e_1 and e_2 but not belonging to either, it is represented by including this common word along with its dependencies for e_1 and e_2. This is usually the case of non-local dependencies. In Example A, it can be illustrated by (*Person–nsubj–obtained–dobj–Degree*).

4.2 Entity Extraction using CRF

The first stage of IE is entity extraction. An entity is defined as a token or a group of tokens which belong to some predefined categories depending on the task. Since our main goal is to extract affiliation relations, we identify six relevant entity types namely *Person*, *Degree*, *University*, *Discipline*, *Organization* and *Position*.

Person and *Organization* entities were classified using Stanford's NER software (Finkel et al., 2005) which makes use of a CRF classifier. For the remaining entities, we train a CRF-based classifier similar to the Stanford's NER, making using of features as described below.

1. Surface tokens (bag of words): For each word token w, all the words in a window size of five, with two words on either side of w are considered. Unigrams, bigrams and trigrams are taken into account.

   In Example A, the surface token features spanning the first five words ("Prof.", "John", "Oliver", "obtained" and "a") are:

   - Unigrams: `Prof.`, `John`, `Oliver`, `obtained`, `a`
   - Bigrams: `(Prof., John)`, `(John, Oliver)`, `(Oliver, obtained)`, `(obtained, a)`
   - Trigrams: `(Prof., John, Oliver)`, `(John, Oliver, obtained)`, `(Oliver, obtained, a)`

2. Part of Speech (POS) Tags: The part of speech for a token like NNP (noun), PRP (pronoun) and IN (preposition) is a strong syntactic feature. For each word token w, POS tags for all the tokens in a window size of five, with two words on either side of w are considered. The POS tags for unigrams, bigrams and trigrams are also taken into account. In Example A, the POS tag features spanning the first five words are:

   - Unigrams: `NNP`, `NNP`, `NNP`, `VBD`, `DT`
   - Bigrams: `(NNP, NNP)`, `(NNP, NNP)`, `(NNP, VBD)`, `(VBD, DT)`
   - Trigrams: `(NNP, NNP, NNP)`, `(NNP, NNP, VBD)`, `(NNP, VBD, DT)`

3. Presence in word list: We have created gazetteers of degrees, positions, disciplines and universities by crawling the web. Presence of a word w in the respective gazetteer indicating a potential entity mention is used as a feature.

   For example: Lemmatized form of degrees `(PhD, BEng, BA, etc.)`, positions `(Professor, Associate Professor, Assistant, etc.)` and Universities with their abbreviations `(University of Melbourne, Unimelb, ANU, etc.)`

We considered all the permutations of these features in an incremental fashion to train CRF models using the scikit-learn toolkit (Pedregosa et al., 2011) as described in Section 6.

4.3 Complex *n*-ary Relation Extraction using SVM

The second stage of IE system is relation extraction. A relation links two or more entities based on predefined rules to render meaningful information. In this work, we are interested in extracting *n*-ary affiliation relations ($n \geq 2$).

We classify each candidate entity pairs or a group of entities within a sentence into three affiliation relation categories namely *binary (2-ary)*, *ternary (3-ary)* and *quaternary (4-ary)* as described in Section 5. We train a SVM with radial basis function (RBF) kernel to classify groups of entities within a sentence using these features:

1. Bag of verbs: All the verbs present in between the entities of a sentence. For example, "obtained", "completed", "graduated".

2. Extracted entities: The entities extracted for each sentence from Stage 1 are strong indicators of presence of a relation. The six entity categories correspond to six different features while training a SVM. If either of the six entity categories is present in a candidate sentence, the corresponding feature is set to 1. Since our entity extraction system is not 100% accurate, there might be some entities in a few sentences which might not be identified correctly. For such instances, we just use the entities which are identified correctly and leave the ones which are not.

   For example: In example A, the entities identified in stage 1 are: (e1, Prof. John Oliver), (e2, Ph.D.), (e3, statistics) and (e4, Stanford University). The entity features corresponding to *Person*, *Degree*, *Discipline* and *University* are set to 1, while the features corresponding to other entity categories remain 0.

3. Part of Speech (POS) sequence: The part of speech sequence connecting the entity type acts as a pattern, the presence of which is used as a feature for the SVM classifier. This feature is important as it makes use of the syntactic structure coupled with the entity information. We observe that many of the POS sequence patterns occur frequently for many documents in our dataset, which rules out the possibility of pattern sparsity.

   In Example A, the POS sequence is (*Person-VBD-DT-Degree-IN-Discipline-DT-University*).

   In cases where an entity is not identified by our entity extractor, we consider the POS tag sequence of the missed entity in lieu of the actual entity type.

   In Example B with *Discipline* not being identified, the POS sequence is (*Person-VBD-DT-Degree-IN-NN-DT-University*).

4. Shortest path dependency information:

   The shortest path dependency based rules are essentially patterns, which act as features for the SVM. This feature is used as described in Section 4.1. The shortest path dependency based rules for each candidate group of entities identified in a given sentence are represented as patterns across all the documents in the corpus. The dependency parse of each candidate sentence is checked for the presence of these patterns. If a pattern is present, the corresponding feature is set to 1.

   For Example A, some of the patterns are: (*Person–nsubj–obtained–dobj–Degree*), (*Person–nsubj–obtained–dobj–Degree–nmod–Discipline*) and (*Person–nsubj–obtained–dobj–Degree–nmod–Discipline–nmod–University*).

   For Example B, some of the patterns are: (*Person–nsubj–completed–dobj–Degree*), (*Person–nsubj–completed–dobj–Degree–nmod–University*)

We considered all the permutations of these features in an incremental fashion to train SVM models using RBF kernel. The predicted tags are compared against the manually annotated gold relation data from AuRes and AuSem datasets described in Section 5. Depending on the number of identified entities (n) within a sentence and the association of these n entities, the relation for a given sentence is categorized into binary, ternary or quaternary relation. We adopted a grid search on C and γ using 10-fold cross validation to prevent overfitting. The experiments are described in Section 6.

5 AuRes and AuSem Corpora

The standard datasets like ACE do not provide annotations for complex *n*-ary relations where $n > 2$. The general affiliation relation category in ACE 2005 dataset contains annotations for only binary relations between entities like *Organization* and *Location*, e.g., <Microsoft, Redmond>. This makes it hard for complex *n*-ary relation extraction where the number of related entities is more than two, which gave rise to the development of two new datasets [1] with annotations for complex relations.

1. AuRes - A collection of 400 documents containing biographical information retrieved from the webpages of researchers and faculty of Australian universities, contains 4092 entities and 1152 relations.

2. AuSem - A collection of 300 seminar announcement mails containing speaker's biography from the department mailing list of the University of Melbourne, contains 2864 entities and 983 relations.

[1] https://github.com/gittykhirbat/nary_datasets

5.1 Label Description

Both AuRes and AuSem are manually annotated with entities and relations following the same annotation guidelines as described below.

5.1.1 Entities

We have identified six different entities which describe the biographical information of a person. We make use of Stanford NER system (Finkel et al., 2005) to classify entities like *Person* and *Organization* as the classification accuracy is very high. For the remaining four entities, we annotate the documents using the following guidelines.

- *Degree*: Token having information related to a degree like B.Sc, PhD, masters or identifiers like undergrad, postrgad, doctoral.
- *University*: Token indicating name of a university or its abbreviation, like "University of Melbourne", "Unimelb", "USyd"
- *Discipline*: Token containing information about a subject or discipline, e.g., Computer Science, Mathematics, Economics.
- *Position*: Token indicating the position of a person in the university of an organization, e.g., Software Engineer, Lecturer, Teacher.

5.1.2 Relations

The documents are annotated for affiliation relations spanning the six entities. The affiliation relation types can be categorized into three classes:

1. *Binary*: When only two entities out of all the identified entities within a sentence are related. For example, in the sentence "Prof. John Oliver did his Ph.D. under the supervision of Prof. Henkel", there are only two entities which satisfy the affiliation relation, <Prof. John Oliver, Ph.D.>.
2. *Ternary*: When three out of all the identified entities within a sentence are related. For example, in the sentence "Prof. John Oliver obtained his Ph.D. in statistics under the supervision of Prof. Henkel", only three entities satisfy the affiliation relation, <Prof. John Oliver, Ph.D., statistics>
3. *Quaternary*: When four out of all the identified entities within a sentence are related. For example, in the sentence "Prof. John Oliver obtained a Ph.D. in statistics from Stanford University under the supervision of Prof. Henkel", four entities satisfy the affiliation relation, <Prof. John Oliver, Ph.D., statistics, Stanford University>

5.2 Annotation

We used Brat annotation tool (Stenetorp et al., 2012) to annotate the document for entities and relations. The annotation task was carried out by two annotators with high proficiency in English. The gold standard was created by detecting annotation overlaps by the two annotators. Legitimate disagreements were resolved by adding an extra attribute to the annotation guidelines which seeks the confidence of annotation on a categorical scale consisting of three values - high, medium and low. The inter-annotator agreement, as computed by Cohen's Kappa measure (Cohen, 1960), was 0.86 for entity annotations and 0.81 for relation annotations.

6 Experiments

6.1 Entity Extraction

For both AuRes and AuSem datasets, we split the data into 70% training and 30% testing datasets. The training data is further split into 90% training and 10% development datasets. The features mentioned in Section 4.2 are employed to train a CRF model using 10-fold cross validation. We train the model in an incremental fashion. Model **M1** makes use of surface tokens which forms baseline for entity extraction. Model **M2** adds POS tag information to M1. Model **M3** adds word list presence feature to M1 and finally model **M4** combines all the features to train the CRF.

These models are used for predictions on the testing dataset, results (F-score in %) for which are shown in Table 1. The best result is obtained when surface tokens, POS tags and presence in word list features are used together. The F-scores for *Person* and *Organization* which are identified using Stanford's NER system are 83.31% and 86.79% respectively.

6.2 *N*-ary Relation Extraction using SVM

We conduct two experiments for relation extraction. First, we run the relation extractor on gold standard entity annotations. This is followed by running the relation extractor on the entities identified by our system in the Stage 1. For both the

Table 1: Entity Extraction Results

Entity	AuRes				AuSem			
	M1	**M2**	**M3**	**M4**	**M1**	**M2**	**M3**	**M4**
Degree	84.85	83.88	85.37	**95.63**	80.31	82.97	84.48	**92.16**
University	79.02	81.27	81.38	**93.88**	78.53	79.92	80.69	**93.33**
Discipline	83.14	91.65	92.22	**92.41**	80.78	86.32	87.18	**88.43**
Position	59.44	61.51	61.02	**93.27**	59.18	60.86	61.19	**89.27**

experiments, we split the data into 70% training and 30% testing datasets. The training dataset is further split into 90% training and 10% testing datasets. We adopted a grid search on C and γ using 10-fold cross validation to prevent overfitting. Pairs of (C, γ) were tried and the one with the best cross-validation accuracy was picked, which in our case turned to be $(2^2, 2^{-3.5})$.

The features mentioned in Section 4.3 are employed incrementally to train a SVM classifier with RBF kernel. The model using bag of words and entity presence features is our baseline system for this task. The SVM models are used for predictions on the testing dataset. Table 2 shows results for both sets of experiments for both the datasets. The columns **Gold** and **Identified** show the results of performing relation extraction using gold standard entity annotations and the system-identified entities respectively. Table 3 gives an account of the performance for extracting binary, ternary and quaternary relations.

7 Discussion

7.1 Error Analysis for Entity Extraction

An account of the entity-wise performance is provided here:

1. *Person*: We used Stanford's NER system for this entity. It was able to classify most of the English names correctly, did well on classifying some non-English names like "Katerina", "Yassaf", "Amit". However, it gave false positives like "Dahab", "Vic" (which are location names); "Rio Tinto", "Leightons" (which are Organization names); "Curtin" (which is a University name); "Dean" (which is a position name) and "Geojournal", "J.J.Immunol." (which are Journal names). These false positives appeared to be a result of the context in which they were being classified. It also resulted in some false negatives like "Cherryl", "Long", "Wai-Kong", which majorly happened because of uncommon names.

2. *Degree*: We used our CRF model to classify *Degree* entities, which performed well mainly due to an extensive gazetteer of most of the degrees which we used as a feature to train the CRF. It can classify degrees and their abbreviations like "Bachelor of Engineering", "B.E.", "BA (Hons.)", "PhD".

3. *University*: Our CRF model performs well in classifying University entities. This is because of a gazetteer of the university names which contains full names of the universities as well as their abbreviations and aliases. e.g., "The University of Melbourne", "Unimelb", "Melbourne Uni". Some of the false negatives arise in documents where the university name is not mentioned conventionally. e.g., "University of WA" (instead of "University of Western Australia" or "UWA").

4. *Organization*: Stanford's NER system is used for this entity. It did well in classifying most of the Organization entities. However, we witnessed some false negatives. It was not able to classify some not so well-known organizations (like "Action Supermarkets", "Freja Hairstyling", "Strategic Wines") and new companies and startups (like "Tesla Motors", "SpaceX").

5. *Position*: A gazetteer of academic positions like "Professor", "Lecturer" was used to classify such positions. However, more specific positions like "Bankwest Professor", "Inaugural Director" and "Founding member" got missed.

6. *Discipline*: Our CRF model was able to classify most of the higher-level disciplines like "Engineering", "Computer Science", "History" based on our gazetteer. However, it

Table 2: Relation Extraction: Comparison of gold standard with system identified entities

Features	AuRes						AuSem					
	Gold			Identified			Gold			Identified		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Bag of words	.64	.59	.62	.57	.53	.55	.59	.54	.56	.54	.48	.51
+ Entity presence (Baseline)	.73	.65	.69	.66	.60	.63	.67	.62	.64	.62	.57	.59
+ POS Tag sequence	.78	.74	.76	.73	.65	.69	.76	.72	.74	.72	.68	.70
+ Shortest path dependency	**.86**	**.82**	**.83**	**.82**	**.73**	**.77**	**.87**	**.82**	**.85**	**.84**	**.73**	**.78**
UPenn System	.76	.71	.73	.66	.73	.69	.76	.73	.74	.65	.74	.69

Table 3: Relation Extraction: Performance across n-ary relations

Features	2-ary			3-ary			4-ary		
	P	R	F1	P	R	F1	P	R	F1
Bag of words	0.61	0.59	0.60	0.58	0.57	0.56	0.52	0.46	0.49
+ Entity presence (Baseline)	0.68	0.64	0.66	0.67	0.61	0.64	0.61	0.55	0.58
+ POS Tag sequence	0.75	0.73	0.74	0.71	0.69	0.70	0.65	0.63	0.64
+ Shortest path dependency	**0.83**	**0.79**	**0.81**	**0.81**	**0.75**	**0.78**	**0.76**	**0.70**	**0.73**
State-of-the-art (UPenn System)	0.74	0.70	0.72	0.71	0.68	0.69	0.69	0.63	0.66

could not classify granular domains within major disciplines like "Equity and Tax", "Shakespearean Literature".

7.2 Error Analysis for Relation Extraction

An account of the n-ary relation extraction system is provided here. Shortest path dependency-based rules prove to be the most effective feature for the trained SVM.

7.2.1 What Worked Well

- Simple relations: Sentences in which the entities are present in a non-complex way. For example, in the sentence "Corinne Fagueret has a Master of Environmental studies completed at Macquarie University", our system extracts <*Person, Degree, University, Discipline*> = <Corinne Fagueret, Master, Macquarie University, Environmental Studies>.

- Complex relations: Sentences in which the entities are present in a non-conventional way. For example, in the sentence "After getting the University of Sydney Science Achievement Prize in 2000 for getting the best weighted average mark for a BSc student, Peter graduated with first class honours and a medal in 2001", our system can extract <*Person, Degree, University*> = <Peter, BSc, University of Sydney>.

- Multiple relations spanning multiple entities: Our system can extract multiple relations from sentences. For example, in the sentence "Angeline is the President of the Lane Cove Bushland and Convener of the better Planning Network", our system can extract <*Person, Position, Organization*> = <Angeline, President, Lane Cove Bushland> and <*Person, Position, Organization*> = <Angeline, Convener, Better Planning Network>.

- Multiple relations spanning same entities: For example, in the sentence "Dr. John Oliver is an Assoc. Prof. and Head in the Department of Finance", our system can extract <*Person, Position, Organization*> = <John Oliver, Assoc. Prof., Department of Finance> and <*Person, Position, Organization*> = <John Oliver, Head, Department of Finance>.

7.2.2 What Did Not Work Well

- Limitation of entity extractor: One bottleneck for our system is the entity extractor sub-system. Even though we have managed to achieve high F-scores for entity extraction, there are cases in which a few entities are missed due to data sparsity. This prohibits the relation extraction. For a given sentence containing n entities, if x entities are identified

by our entity extraction sub-system then our relation extraction sub-system makes use of the features to learn valid subset of relations occurring among the $n - x$ entities.

- Limitation of parser: Our system faces ambiguity in cases where an appositive dependency occurs between two entities. For example, in the sentence "Associate Professor Christoff Pforr (PhD) is Course Coordinator for Tourism and Hospitality and Group Leader of the Research Focus Area Sustainable and Health Tourism with the School of Marketing, Curtin Business School", School of Marketing and Curtin Business School are both classified as University entities with an appositive relation between the two because of the common word "School". While extracting relation, it is not clear which entity should be considered.

- Ambiguity in choosing correct entity: Sentences containing multiple entities with the same context cause an ambiguity. For example, in the sentence "Sarah is currently co-investigator with Professor Fiona Haslam for a study commissioned by Rio Tinto through the University of Adelaide". In this sentence, there are two associations for Sarah - Rio Tinto and University of Adelaide. The system renders both, giving us a false positive <Sarah, co-investigator, Rio Tinto>.

- Unknown words from other language: For example, in the sentence "Marios holds a PhD in Political Science from Northern Territory University and a Staatsexamen in Geography and Political Science as well as a Teaching Certificate from the University of Tübingen (Germany). Staatsexamen and Tübingen are not detected, thereby causing errors.

- Inference-based relations: Inference of relation from previous sentences in the paragraph can not be done as our system lacks long distance dependency information. For example, in the sentence "Ruhul words as a tutor for Biotechnology at RMIT University. He also worked in a similar position at the University of Melbourne.", we are unable to infer what "similar position" mean. This would be explored in the future.

7.3 Comparison with other state-of-the-art IE systems

A comparison with the UPenn system (McDonald et al., 2005) is provided in Table 2 and 3. We re-implement this system and train it on our training and development datasets using 10-fold cross validation. The learnt system is used to predict the relations for testing dataset. At the time of this work, this system is the state-of-the-art in complex *n*-ary relation extraction, with an F1-score of 69.42% on a dataset of 447 abstracts selected from MEDLINE. On our datasets of AuRes and AuSem, their technique achieved F1-Score of 69.44% and 69.22% respectively as compared to 77.49% and 78.38% respectively using shortest path dependency based rules, which shows an improvement of 8% F1-score. Our technique obtained far less false positives and a comparable recall.

8 Conclusions and Future Work

Through this paper, we show a new approach to *n*-ary relation extraction using shortest path dependency based rules which provides an improvement of 8% F1-score over the state-of-the-art. Two stage extraction procedure involving CRF-based entity extraction and SVM-based relation extraction is proposed to extract affiliation relations. An empirical analysis is conducted over two manually annotated datasets to validate this method. The manually annotated datasets could be used for the advancement of natural language processing research in the future.

For future work, it would be interesting to investigate the usage of shortest path parse tree for *n*-ary relation extraction since sentence parsing provides a semantically rich information about a sentence. It would also be interesting to explore *n*-ary relation extraction spanning across multiple sentences. Finally, future use of the introduced corpora in research to augment existing knowledge bases could yield interesting insights.

Acknowledgments

Jianzhong Qi is supported by the Melbourne School of Engineering Early Career Researcher Grant (project reference number 4180-E55), and the University of Melbourne Early Career Researcher Grant (project number 603049).

References

Enrique Alfonseca and Suresh Manandhar. 2002. An unsupervised method for general named entity recognition and automated concept discovery. In *In: Proceedings of the 1 st International Conference on General WordNet*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: A high-performance learning name-finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, ANLC '97, pages 194–201, Stroudsburg, PA, USA. Association for Computational Linguistics.

Razvan C. Bunescu and Raymond J. Mooney. 2005a. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA. Association for Computational Linguistics.

Razvan C. Bunescu and Raymond J. Mooney. 2005b. Subsequence kernels for relation extraction. In *Proceedings of the 19th Conference on Neural Information Processing Systems (NIPS)*. Vancouver, BC, December.

J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. A framework and graphical development environment for robust nlp tools and applications. In *ACL*, pages 168–175.

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*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, ACLdemo '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hong Li, Sebastian Krause, Feiyu Xu, Andrea Moro, Hans Uszkoreit, and Roberto Navigli. 2015. Improvement of n-ary relation extraction by adding lexical semantics to distant-supervision rule learning. In *ICAART 2015 - Proceedings of the 7th International Conference on Agents and Artificial Intelligence*. SciTePress.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. 2005. Simple algorithms for complex relation extraction with applications to biomedical ie. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 491–498, Stroudsburg, PA, USA. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: A web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 102–107, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106, March.

Shubin Zhao and Ralph Grishman. 2005. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 419–426, Stroudsburg, PA, USA. Association for Computational Linguistics.