Ontology Learning from Twitter Data

Saad Alajlan[1,2], Frans Coenen[1], Boris Konev[1] and Angrosh Mandya[1]
[1]*Department of Computer Science, The University of Liverpool, Liverpool, U.K.*
[2]*College of Computer and Information Sciences, Al Imam Mohammad Ibn Saud Islamic University, Riyadh, Saudi Arabia*

Keywords: Ontology Learning, RDF, Relation Extraction, Twitter, Name Entity Recognition, Regular Expression.

Abstract: This paper presents and compares three mechanisms for learning an ontology describing a domain of discoursed as defined in a collection of tweets. The task in part involves the identification of entities and relations in the free text data, which can then be used to produce a set of RDF triples from which an ontology can be generated. The first mechanism is therefore founded on the Stanford CoreNLP Toolkit.; in particular the Named Entity Recognition and Relation Extraction mechanisms that come with this tool kit. The second is founded on the GATE General Architecture for Text Engineering which provides an alternative mechanism for relation extraction from text. Both require a substantial amount of training data. To reduce the training data requirement the third mechanism is founded on the concept of Regular Expressions extracted from a training data "seed set". Although the third mechanism still requires training data the amount of training data is significantly reduced without adversely affecting the quality of the ontologies generated.

1 INTRODUCTION

Social media data provides a wealth of information that can be tapped to generate actionable knowledge. There have been a number of studies where social media data has been successfully employed for prediction purposes, for example the outcomes of elections (Murthy, 2015) or flu outbreaks (Aramaki et al., 2011). However, there have been few studies directed at facilitating the querying of social media data for information retrieval purposes. The principal challenges arises from the unstructured nature of the data, which makes it difficult to utilise for data querying purposes. What is required is a general purpose ontology which can be used to impose structure on unstructured social media data. However desirable such an ontology might be, a global ontology that covers every "domain of discourse", whether featured in social media or not, is currently beyond the means of computer science; although it should be acknowledged that a great many domain specific ontologies have been generated, especially in the context of semantic web services (Klusch et al., 2016). What can be done is to use domain specific ontologies; typically an enquirer will only be interested in some specific social media domain of discourse. Where these ontologies exist, well and good; however, where they do no exist they will need to be generated. Ontology generation is a resource intensive undertaking; the key challenge is in identifying and defining the various entities and relations that represent the target domain and need to be included in the desired ontology. A solution is to automate the process by employing some form of ontology learning. Ontology learning, also known as ontology extraction, ontology generation or ontology acquisition, is concerned with the automatic or semi-automatic creation of ontologies (Zhou, 2007).

The idea proposed in this paper is to use ontology learning to identify the entities and corresponding relations between entities, from a corpus of social media data texts, and then use this to define a domain specific ontology. The focus for the work is Twitter data, because: (i) it is readily available, (ii) specific domains can be simply defined using "# tags" and (iii) the desired ontologies are limited (comprising a small number of entities and relations). Thus, more specifically, given a collection of tweets from within a particular Twitter domain, the idea is to use Natural Language Processing (NLP) tools and techniques (King and Reinold, 2014) to identify entities, and relationships between entities, in the Twitter data collection and then use this to define an ontology spanning this collection, expressed using an ontology language that facilitates information retrieval. The paper presents and compares three mechanisms for NLP-based ontology learning in the context of social me-

Alajlan, S., Coenen, F., Konev, B. and Mandya, A.
Ontology Learning from Twitter Data.
DOI: 10.5220/0008067600940103
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 94-103
ISBN: 978-989-758-382-7

dia (Twitter) data querying: (i) using the Stanford CoreNLP toolkit (Finkel et al., 2007; Chunxiao et al., 2007), (ii) using the General Architecture for Text Engineering (GATE) toolkit (Cunningham, 2002) and (iii) using Regular Expressions (Sidhu and Prasanna, 2001) coupled with CoreNLP. The adopted ontology language was the Resource Description Framework (RDF) (Graham and Carroll, 2004) which readily supports querying.

The entity and relation extraction mechanisms were evaluated using standard machine learning metrics coupled with ten-fold cross validation. The generated ontologies were evaluated by examining the syntax of the RDF using a "validator" tool recommended by the World Wide Web Consortium (W3C) and visual inspection of the semantics. The information extraction utility of the populated ontologies was evaluated by directing SPARQL queries at the populated ontologies; if the results obtained from the SPARQL querying were correct, it could be claimed that the proposed approaches served their purpose.

The rest of this paper is structured as follows. Section 2 gives a brief overview of previous work on ontology learning. Sections 3, 4 and 5 then present the three proposed mechanisms for ontology learning from Twitter data. An evaluation and comparison of the three proposed techniques is presented in Section 6. Some conclusions are presented in Section 7.

2 PREVIOUS WORK

To the best knowledge of the authors there has been no specific work on ontology learning from Twitter data. However, there has been previous work on ontology learning from free text. Parallels can therefore be drawn between the previous work on ontology learning from free text and Twitter data; the distinction is that Twitter records are typically shorter than free text records and are typically less well formed. A brief review of the previous work on free text ontology learning is thus presented in Sub-section 2.1. There are a number of reports where the relation extraction from text has been automated, or semi automated. Relation extraction is an important element of the proposed approaches; automated entity and relation extraction is the first step to automate the ontology learning process. A review concerning previous work on automated relation extraction is therefore presented in Sub-section 2.2.

2.1 Ontology Learning

Examples of mechanisms for ontology learning from free text can be found in (Exner and Nugues, 2012) and (Republic, 2003). In (Exner and Nugues, 2012), a system was described to automatically extract triples from unstructured data, supported by DBpedia, so as to generate ontology classes. The system operated using a semantic parser and a co-reference solver, and used an ontology base-mapping system that utilized DBpedia to infer relations between the entities. DBpedia was also used to support the identification of ontology classes. The evaluation was conducted manually by analysing 200 randomly selected sentences, the F-score with respect to the mapped triples was 66.3%. The limitation is that the approach is restricted to the content of wikipedia.

In (Republic, 2003) a mechanism that used Text-to-Onto to extract pairs of terms based on the TF-IDF (Term Frequency - Inverse Document Frequency) measure was described. After extracting pairs of terms, a part of speech tagger was used to discover verbs, which were considered to describe relationships between the pairs of terms according to the frequency of the co-occurrence of the verbs and entity pairs. All pairs were mapped to concepts using the TAP knowledge base, developed at Stanford. TAP is a large repository of lexical entries, such as proper names of places, companies, people, but also names of sports, art styles and other less traditional named entities. However, the approach is limited to what is available within TAP.

2.2 Relation Extraction

Many reports present methods for automating the relation extraction from text process using machine learning techniques, specifically supervised learning (Carlson et al., 2010; Riedel et al., 2010). Of particular relevance with respect to the work presented in this paper is work where the Stanford CoreNLP tool and GATE have been used for relation extraction (Chunxiao et al., 2007; Wang et al., 2006). In (Chunxiao et al., 2007) a supervised information extraction system was introduced, founded on the Stanford CoreNLP tool, which could be customised. The domain considered in (Chunxiao et al., 2007) was the USA National Football League (NFL) Scoring corpus. The corpus contains 110 article relating to NFL. In (Wang et al., 2006), a Support Vector Machine (SVM) model was used to perform multi-class relation classification by using a sequence of SVM binary classifiers (the one-against-one method). The GATE tool was used to define the Machine Learning (ML)

features by using tokenisation, sentence splitting, part of speech tagging and noun and verb phrase chunking. The authors of (Wang et al., 2006) also used WordNet to provide word sense disambiguation. A particular challenge of supervised learning for relation extraction from text is the need for a training sets which are usually manually generated (Riedel and Mccallum, 2013; Takamatsu et al., 2012; Carlson et al., 2010). This is a criticism that can also be directed at the Stanford and GATE-based methods presented later in this paper. One solution is to use some form of semi-supervised learning. One example can be found in (Carlson et al., 2010) where an iterative training method, directed at web page free text, was presented that involved "self-supervision". The process presented in (Carlson et al., 2010) commences with small amount of labeled training data to train an initial classifier which is then used to iteratively label further training data. The third approach presented in this paper, the regular expression-based approach, also adopts a semi-supervised approach. An interesting relation extraction approach is presented in (Riedel et al., 2010) where a system is described that avoids using labelling training data by using an external knowledge base instead (namely Freebase). However, in the case of the Twitter domain of interest with respect to this paper it was expected that no such knowledge-base would be available (although with respect to some domains of discourse this might be the case).

3 ONTOLOGY LEARNING USING STANFORD CORE NLP

In this and the following two section the three mechanisms for extracting ontologies from Twitter data considered in this paper are presented, commencing with the Stanford Core NLP approach. The pipeline architecture for the Stanford ontology learning framework is given in Figure 1. From the figure it can be seen that the process starts with a collection of tweets T. The twitter data is then cleaned (not shown in the Figure), for example by deleting hyperlinks. The next stage is the knowledge extraction stage which comprises: (i) Named Entity Recognition (NER) and (ii) Relation Extraction. The next stage is mapping the identified entities to classes; for example, the class "countries" which includes objects such as UK, USA and China. Then the classes and identified relations are used for ontology generation. The result is a RDF represented ontology. More details concerning the NER and relation extraction models, and the ontology generation step, are provided in the following three sub-sections.

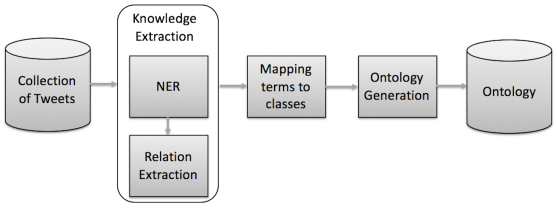

Figure 1: Stanford Ontology Learning Framework.

3.1 Named Entity Recognition (NER)

The primary objective of the NER model is to identify the entities that feature within the Twitter collection (after which they will be associated with classes). By default the Stanford NER model will identify entities belonging to seven different "standard" classes: (i) Location, (ii) Person, (iii) Organisation, (iv) Money, (v) Percent, (vi) Date and (vii) Time. Although the Stanford NER model is reasonably good at identifying entities belonging to these standard classes any Twitter domain of discourse cannot be expected to adhere to these standard classes. The model therefore needs to be retrained to take into account the other entity classes that feature in a given domain of discourse. For example, given the "motor vehicle pollution" domain of discourse considered for evaluation purposes later in this paper the generated ontology should reflect the environmental hazards of vehicles and include entities such as "petrol car" and "diesel car". In order to identify such entities the Stanford NER tool provides the means whereby the model can be retrained given an appropriately constructed training set where the entities of interest have been annotated. Figure 2 shows an example training record, in the syntactical format required by the NER tool, that may be used to create a model to identify the entities belonging to the classes: Location, Date and Fuel vehicles. The example given in the figure expresses the tweet "Norway to completely ban petrol powered cars by 2025", where: (i) the label "Loc" indicates that the associated word belongs to the class Location, (ii) the label "O" indicates a *wild card*, (iii) the label "FuelV" indicates a word (entity) belonging to the class Fuel Vehicles and (iv) "Date" an entity to be associated with the class Date. The NER model is used twice in the Stanford ontology learning framework. Firstly to associate entities with classes (as described in this sub-section), and secondly as a part of the relation extraction tool (described below) to identify relations between entities.

```
Norway  Loc
to      0
`       0
completely      0
ban     0
petrol  FuelV
powered FuelV
cars    0
by      0
2025    Date
'       0
```

Figure 2: Example Stanford NER training record.

3.2 Relation Extraction

Once an appropriate NER model has been created the Stanford relation extraction tool can be used to create the required relation extraction model. As in the case of the NER Model the Stanford relation extraction tool includes the means whereby a relation exaction model can be trained using an appropriately defined training set (Roth and Yih, 2019). This was also the approach used in (Chunxiao et al., 2007) where Stanford relation extraction was used to identify and extract relations in the domain of American football, although not from Twitter data. The training data needs to highlight entities and the relations between them. In the proposed process the entities were identified using the generated NER model (see above). An example training record is given in Figure 3. As in the case of the entity example given in Figure 2, the example uses the tweet "Norway to completely ban petrol powered cars by 2025". The Part of Speech (PoS) tag is given in column 5 and the content of the tweet in column 6. The example expresses two relations: (i) the relation "ban" that exists between word 0 and word 5 (entities "Norway" and "petrol powered"), and (ii) the relation "Ban fuelV Date" that exists between words 0 and 7 (entities "Norway" and "2025").

The relation model, once trained, was used to extract entities and relations from a given Twitter data set. The way that the Stanford relation extraction model operates means that additional relations may be identified that are not pertinent to the domain of discourse. The results were therefore filtered so that only the relations identified in the training data were retained. The filtered results were then stored as a set of triples of the form $\langle entity1, relation, entity2 \rangle$. The results with respect to the example Tweet "Norway to completely ban petrol powered cars by 2025" will therefore be the triples $\langle Norway, ban, petrol\ powered \rangle$ and $\langle Norway, ban\ fuelv\ Date, 2025 \rangle$.

3.3 Ontology Generation

The final step in the ontology learning frameworks given in Figure 1 was the ontology generation step. There are many tools recommended by The World Wide Web Consortium (W3C) to convert entity-relation-entity triples to an RDF represented ontology depending on the nature of the data. In this research LODRefine was used, which is the OpenRefine tool with an RDF extension (Harlow, 2015). Figure 4 shows a simple example of an RDF ontology for the motor vehicle pollution scenario.

4 GATE ONTOLOGY LEARNING FRAMEWORK

The GATE-based approach to ontology learning is presented in this section. An overview of the framework is given in Figure 5. Inspection of Figure 5 and Figure 1 indicates that the distinction between the two is in the Knowledge Extraction step. Note that for this purpose Knowledge Extraction two GATE components are used, the Gazetteer and the Relation Extraction components. The gazetteer uses a prescribed lists of words, describing entity classes, and uses these lists of words to identify entities within given texts. The relation extraction component is then used to identify relations that exist between pairs of entities. The last two steps are the same as those described for the Stanford Framework described above.

In more detail, the Knowledge Extraction process was as follows: (i) data preprocessing, (ii) entity extraction using the gazetteer, (iii) class pairing (iv) training set generation, (v) relation extraction model generation and (vi) relation extraction model application. The data preprocessing comprised the application of a number of NLP pre-processing to the Twitter data, namely word tokenisation and part of speech tagging. This pre-processing steps was done using ANNIE (A Nearly New Information Extraction system) tool available within GATE. ANNIE assigns a sequential character ID number, c_i, to each character in a a given Tweet T, $T = [c_1, c_2, \ldots]$. Each word is therefore defined by a start and end character; a word ID is thus expressed as $\langle c_s, c_e \rangle$; this is illustrated in Figure 6 where a word annotated Tweet is given. The next step was to use the Gazetteer to identify, and annotate, the words in a Twitter data collection, so as to identify entities that exist in the data and consequently assign class labels to those entities. GATE comes with a number of predefined gazetteer files (lists), such as locations, organisations and dates; but for the proposed ontology generation from Twitter data applica-

```
0       Loc     0       0       NNP     Norway  0       0       0
0       0       1       0       TO      to      0       0       0
0       0       2       0       ``      `       0       0       0
0       0       3       0       RB      completely      0       0       0
0       0       4       0       NN      ban     0       0       0
0       0ther   5       0       NN/VBD  petrol/powered  0       0       0
0       0       6       0       NNS     cars    0       0       0
0       0       7       0       IN      by      0       0       0
0       0ther   8       0       CD      2025    0       0       0
0       0       9       0       POS     '       0       0       0
0       0       10      0       .       .       0       0       0

0       5       ban
0       8       ban_fuelV_Date
```

Figure 3: Example of Stanford relation extraction training data record.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
      xmlns:a="http://www.example.com/pc#"
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

<rdf:Description rdf:about="http://www.example.com/pc#Loc">
      <a:ban rdf:resource="http://www.example.com/pc#FuelV"/>
      <a:use rdf:resource="http://www.example.com/pc#GreenV"/>
      <a:ban_fuelV_Date rdf:resource="http://www.example.com/pc#Date"/>
      <a:use_greenV_Date rdf:resource="http://www.example.com/pc#Date"/>
</rdf:Description>

</rdf:RDF>
```

Figure 4: Example RDF file.

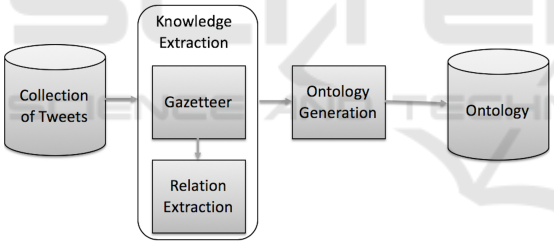

Figure 5: GATE Ontology Learning Framework.

tion the assumption was that appropriate files would not be available in all cases. These must therefore be generated. In the context of the motor vehicle pollution scenario used for evaluation purposes, as noted above, the focus was on four specific classes: (i) Location ("Loc"), (ii) Date ("Date"), (iii) Fuel Vehicle ("Fuelv") and (iv) Green Vehicle ("Greenv"). Using the annotated tweets, JAPE (H. Cunningham D. Maynard and V. Tablan, 2000) was used to pair the classes together. The idea was to link every pair of classes that feature in a Tweet. Gazetteer assigns a unique Entity ID, e, to each identified entity which in turn references a sequence of characters in T.

In GATE, as in the case of Stanford CoreNLP, relation extraction was conducted using a supervised learning approach. This in turn required a training set (the motor vehicle pollution domain of discourse was used for evaluation purposes, which will presented later in this paper). Therefore, once all classes pairings had been identified, the next step was for the user to define a training set by manually assigning relations to each identified pair of classes. This training set was then used to generate a GATE relation extraction model designed to predict classes and relations between them. An example of a GATE relation extraction training record is given in Figure 6. As before, the example is derived from the Tweet "Norway to completely ban petrol powered cars by 2025". The example defines a relation "ban" that links an antecedent entity belonging to the class "Loc" to a consequent entity belonging to the class "Fuelv". In the example, the phrase to be considered is delimited by the charcte ID numbers 0 to 45 where 0 marks the start of the antecedent entity and 45 marks the end of the consequent entity. The specific entities referenced in the example have the Entity ID numbers 26 and 43, which in this case identify the entities Norway ($\langle 0,6 \rangle$) and petrol powered cars ($\langle 26,45 \rangle$) belonging to the classes "Loc" and "Fuelv" respectively. Once the GATE relation extraction model has been trained, the model can be used to predict classes and their relations from tweets. The results were stored as a XML file, $\langle class1, class2, relation \rangle$. Figure 7 gives a GATE relation extraction result for the Tweet "Norway to

```
<TextWithNodes><Node id="0" />Norway<Node id="6" /> <Node id="7" />to<Node id="9" /> <Node id="10" />'<Node id="11" />completely<Node
id="21" /> <Node id="22" />ban<Node id="25" /> <Node id="26" />petrol<Node id="32" /> <Node id="33" />powered<Node id="40" /> <Node
id="41" />cars<Node id="45" /> <Node id="46" />by<Node id="48" /> <Node id="49" />2025<Node id="53" />'<Node id="54" />.<Node id="55" />
></TextWithNodes>


<Annotation Id="1111" Type="RelationClass" StartNode="0" EndNode="45">
<Feature>
  <Name className="java.lang.String">rel-type</Name>
  <Value className="java.lang.String">ban</Value>
</Feature>
<Feature>
  <Name className="java.lang.String">loc</Name>
  <Value className="java.lang.String">26</Value>
</Feature>
<Feature>
  <Name className="java.lang.String">fuelv</Name>
  <Value className="java.lang.String">43</Value>
</Feature>
</Annotation>
```

Figure 6: Example of relation extraction training data record.

```
<AnnotationSet Name="ML">
<Annotation Id="1158" Type="RelationClass" StartNode="0" EndNode="45">
<Feature>
  <Name className="java.lang.String">fuelv</Name>
  <Value className="java.lang.String">43</Value>
</Feature>
<Feature>
  <Name className="java.lang.String">rel-type</Name>
  <Value className="java.lang.String">ban</Value>
</Feature>
<Feature>
  <Name className="java.lang.String">loc</Name>
  <Value className="java.lang.String">26</Value>
</Feature>
<Feature>
  <Name className="java.lang.String">prob</Name>
  <Value className="java.lang.Float">0.92415094</Value>
</Feature>
</Annotation>
```

Figure 7: Example of a GATE Relation Extraction Result.

completely ban petrol powered cars by 2025"; note that an accuracy probability is given.

5 ONTOLOGY LEARNING USING REGULAR EXPRESSIONS (AND STANFORD CORENLP)

While the above described methods (using the Stanford coreNLP and GATE frameworks) provide useful mechanisms for supporting ontology learning, both involve significant end-user resource, particularly in the preparation of training data. The entire process is therefore time consuming and does not generalise over all potential domains. The third mechanism considered in this paper was designed to address the training data preparation overhead by using regular expressions in order to limit the resource required with respect to the previous two frameworks. An overview of the ontology learning using a regular expressions framework is given in Figure 8. Note that the framework interfaces elements of Stanford CoreNLP, it could equally well be interfaced with GATE, however preliminary evaluation (reported on in Section 6 below) indicated that Stanford was a better option.

From Figure 8, the process commences with a collection of tweets T. The first step is to generate an entity annotated with class labels "Seed Set"; in the context of the evaluation presented later in this

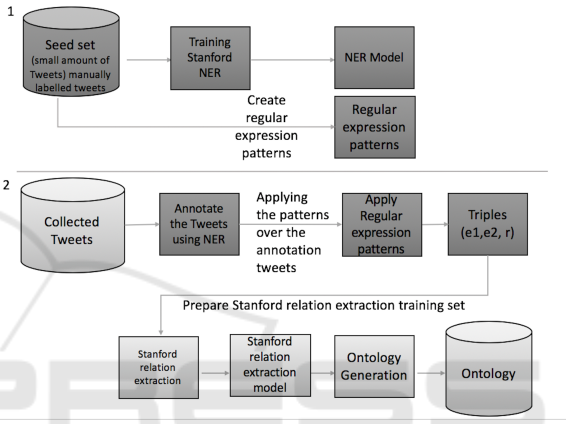

Figure 8: Regular expressions ontology learning framework.

paper 100 tweets were selected instead of the 300 used to evaluate to Stanford and GATE frameworks. This seed set is then used to learn a Stanford NER model in a similar manner as described previously in Sub-section 3.1. The seed set is used to generate a set of regular expressions (patterns). Three catagories of regular expression were considered: (i) two entity expressions, (ii) three entity expressions and (iii) four entity expressions. A entity expressions takes the form $\{e_1, ?, r, ?, e_2\}$, $\{e_1, ?, e_2, ?, r, ?, e_3\}$ and $\{e_1, ?, e_2, ?, e_3, ?, r, ?, e_4\}$ respectively, where ? indicates a random number of intervening words. In each case there are a number of variations, 6, 24 and 120 respectively. Note that the proposed framework offers the advantage, unlike comparable frameworks, that it can operate with more than two entities.

The entire set of tweets T are then annotated using Stanford NER model. Then, the regular expression are applied to the annotated tweets and a set of triples extracted of the form $\langle e_1, e_2, r \rangle$. In some cases several such triples will be extracted from a single Tweet, in other cases no triples will be identified.

The triples are then use to automatically generate a training set for Relation Extraction model genera-

tion as described earlier in Sub-section 3.2. The remaining two steps are identical to those included in the previous two approaches.

6 EVALUATION

This section reports on the evaluation conducted to evaluate the proposed ontology learning frameworks presented in the foregoing three sections. For the evaluation a Twitter dataset, directed at the car pollution domain, was generated. Further details of this data set are presented in Sub-section 6.1. The objectives of the evaluations were:

1. To determine and compare the effectiveness of the Stanford and GATE ontology learning frameworks.
2. To determine the effectiveness of the Regular Expression ontology learning framework.
3. To evaluate the syntactic integrity of the generated ontologies.
4. To evaluate the utility of the generated ontologies.

Each is discussed in further detail in Sub-sections 6.2 to 6.5 below.

6.1 Evaluation Data

The section briefly describes the car pollution domain Twitter data set used for evaluation purposes. The tweets were collected using the Twitter API. The car pollution topic was chosen since it was easy to understand and hence any proposed mechanism using this data could be readilly analysed. The criteria for the collected tweets was that they should contain content related to banning fuel vehicles or using green vehicles in a country or city. 300 tweets were collected and labelled for training purposes. The data set featured four entity classes: (i) *Loc*, (ii) *fuelV*, (iii) *greenV* and (iv) *Date*; and four relations: (i) *ban*, (ii) *use*, (iii) *ban fuelV Date* and (iv) *use greenV Date*. The distribution of the entity and relation classes across the data set is presented in Figures 9 and 10; 768, 384, 198 and 1162 for the entity classes; and 241, 125, 166 and 87 for the relations. Inspection of the figures indicates that there were many more examples of entities than relations. A typical tweet included eight entities and two relations; not all entities were paired. Note also that the number of examples per class was very imbalanced. A further 313 tweets were collected from the same domain; and used to create and populate the ontologies using the learnt models.

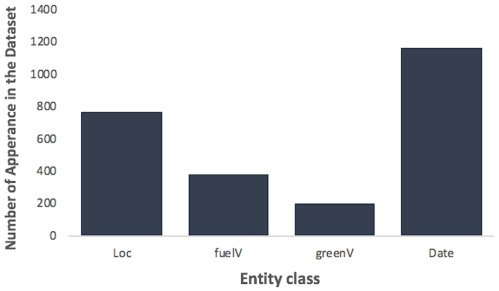

Figure 9: Distribution of the entity classes across the data set.

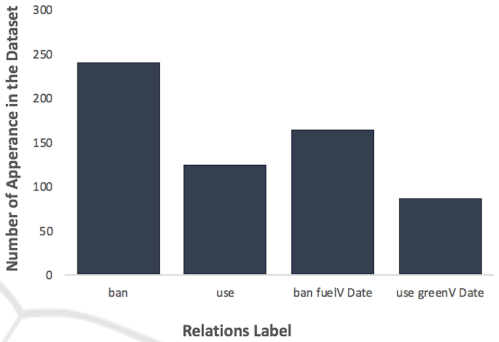

Figure 10: Distribution of relations across the training dataset.

6.2 Effectiveness of the Stanford and Gate Ontology Learning Frameworks

The entire initial data set of 300 tweets was used to generate and evaluate the Stanford NER and Relation Extraction models, and the Gate Relation Extraction model, used to produce ontologies in the context of the proposed Stanford and Gate ontology learning frameworks. Using Stanford relation extraction, the NER model was trained using the four entity classes and the Relation Extraction model using the four relations listed above. Using GATE a gazetteer dictionary was used. Ten-fold Cross Validation (TCV) was used for the evaluation; 270 records for training and 30 for testing. The metrics used were Precision (P), recall (R) and F-score (F).

The results are presented in Tables 1, 2 and 3. Table 1 presents the result obtained with respect to the generated Stanford NER model. Tables 2 and 3 present the results obtained with respect to Stanford and GATE generated Relation Extraction models respectively. Note that precision and recall were not included in Table1 because the Stanford NER tool does not provide these. Inspection of Table 1 shows that a small Standard Deviation (Stand. Dev.) was recorded. It can thus be argued that the generated entity model

Table 1: TCV results for the Stanford NER model evaluation.

Fold Num.	F
1	77.3
2	77.7
3	77.2
4	79.3
5	77.2
6	78.0
7	78.0
8	78.9
9	78.4
10	78.3
Average	78.03
Stand. Dev.	0.71

Table 2: TCV results for the Stanford Relation Extraction model evaluation.

Fold Num.	P	R	F
1	67.6	88.5	76.7
2	81.8	88.5	85.0
3	81.6	75.5	78.4
4	94.2	98.0	96.1
5	47.2	69.0	56.1
6	85.2	74.2	79.3
7	74.3	76.4	75.3
8	70.2	75.5	72.7
9	88.6	87.3	87.9
10	86.7	89.7	88.1
Average	77.74	82.26	79.56
Stand. Dev.	13.58	9.26	10.91

Table 3: TCV results for the the GATE Relation Extraction model evaluation.

Fold Num.	P	R	F
1	73.8	68.2	70.6
2	85.9	77.0	79.4
3	69.2	86.7	71.1
4	50.5	70.2	57.6
5	48.6	60.5	50.8
6	75.2	93.7	82.1
7	53.9	72.2	61.4
8	69.3	75.0	70.0
9	70.7	77.8	72.0
10	86.7	68.8	64.4
Average	68.38	75.01	67.94
Stand. Dev.	13.52	9.57	9.58

was consistent and reasonably accurate and that consequently the mechanism for generating it was effective.

Inspection of Tables 2 and 3 indicates a wide spread of results (high standard deviations) in both cases. The conjectured reason for this was the imbalanced nature of the training data. Inspection of the Fold 5 test data, the worst performing fold, revealed that it included nine examples of the *use greenV Date* relation class but that only two were classified correctly. From Figure 10 it can be seen that there were only 87 examples of the *use greenV Date* class, approximately nine per fold. On the other hand, for the class *ban* there were 241 examples. Comparing Tables 2 and 3 it can be seen that the Stanford modeld produced a better average F-score than the GATE model. This is why it was decided to interface the Regular Expression approach with the Stanford NLP tool as opposed to the GATE tool.

6.3 Effectiveness of the Regular Expression Ontology Learning Framework

For the evaluation of the Regular Expression Ontology Learning Framework a seed training set of 100 records, a third of the data available was used. Recall that for the evaluation of the Stanford and GATE Ontology Learning Frameworks, training sets numbering 270 tweets were used. Once the regular expressions had been identified they were applied to the remaining 200 records. In the context of the Stanford NER model, integral to the Regular Expression ontology learning framework, the evaluation was conducted using three-fold Cross Validation; three because of the size of the seed set. The results are given in Table 4.

Table 4: 3 Fold Cross Validation results for the Regular Expression Stanford NER model evaluation.

Fold Num.	F
1	49.0
2	53.0
3	55.0
Average	52.33
Stand. Dev.	3.06

From the Table it can be seen that the average F-score was less than that obtained using the Stanford "stand alone" framework trained using all 270 records (see Table 1), but within acceptable limits.

Table 5 gives the results obtained with respect to the resulting Relation Extraction model. From the table, it can be seen that the F-score values ranged between 56.9 and 88.5, again because of the imbalanced nature of the training data. However what is interesting to note is that the average Relation Extraction F-score obtained using the Regular Expression approach was better than the GATE approach al-

Figure 11: Ontology graphs.

Table 5: TCV results for the Regular Expression Stanford Relation Extraction model evaluation.

Fold Num.	P	R	F
1	75.9	80.4	78.1
2	89.3	87.7	88.5
3	75.9	63.8	69.3
4	73.8	77.5	75.6
5	70.0	77.8	73.7
6	68.4	72.2	70.3
7	79.6	68.3	73.5
8	75.0	82.4	78.5
9	47.5	70.7	56.9
10	87.0	75.8	81.0
Average	74.24	75.66	74.54
Stand. Dev.	11.51	7.09	8.33

though not as good at the Stanford approach; whilst using a much smaller training set.

6.4 Generated Ontology Evaluation

The generated ontologies were of the form presented earlier in Figure 4 using RDF notation. From inspection of the figure, it can be seen that the semantics of the generated ontology seem correct based on the classes and relations extracted from the tweets. Other than visual confirmation, the generated ontologies were automatically evaluated further by checking their syntactic integrity. This validation was done using the RDF W3C Validation Tool. Figure 11 shows the ontology graph generated using the W3C Validation Tool. Thus it can be concluded that, at least in this simple case, the generated ontologies was semantically and syntactically correct.

6.5 Evaluation of the Utility of the Generated Ontology

To evaluate the utility of the generated ontology, the idea was to populate the ontologies using 313 tweets and then use SPARQL to query the data. The motivation for this was from (Prud'Hommeaux et al., 2008) where it was noted that "SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware". For the evaluation Apache Jena was used to query the generated RDF, one of the recommended tools from W3C. Example SPARQL queries are given in Figure 12. The example queries are directed at identifying all locations that ban any type of car and when the UK will ban fuel vehicles. Thus, from the SPARQL query result, it could be concluded that the generated ontology was appropriate.

```
SELECT ?Location ?vehicle_type
WHERE{
?Location a:ban ?vehicle_type
}
```

```
SELECT ?Date
WHERE{
a:UK a:ban_fuelV_Date ?Date
}
```

Figure 12: Examples of SPARQL query.

7 CONCLUSION

This paper has presented three mechanisms for learning ontologies from Twitter data: (i) the Stanford ontology learning framework, (ii) the GATE ontology learning framework and (iii) the Regular Expression ontology learning framework, the latter coupled with the Stanford NLPcore toolkit. The output from all three mechanisms was an RDF represented ontology generated using LODRefine, a W3C recommended tool. The first two mechanisms required substantial amounts of training data. This presented a significant disadvantage as the preparation of the required training data required considerable end user resource. The third mechanism was designed to address this disadvantage by using a much smaller "seed set" from

which a more complete training set could be generated and input to either of the two previous mechanisms. All three mechanisms were compared using a car pollution scenario comprised of 300 tweets as the training set and 313 tweets to create the ontologies. For the first two mechanisms the Relation Extraction models were compared and Stanford Relation Extraction found to outperform GATE relation extraction; an average F-score of 79.56 compared to 67.94. The third mechanism, the Regular Expression mechanism, was therefore coupled with Stanford Relation Extraction. For the first two mechanisms 270 records were used for training (30 held back for testing) while for the third only 100 records were used fort the initial seed training set, a reduction of 65%, without adversely affecting the quality of the generated ontologies. The generated ontologies were evaluated using a W3C validation tool which focused on the syntax of the ontology, whilst the utility of the ontologies was evaluated by populating the ontologies and querying them using SPARQL test queries. The results were very encouraging and the authors are now embarking on a large scale evaluation of the proposed mechanisms directed at more sophisticated ontologies and using larger collections of tweets.

REFERENCES

Aramaki, E., Maskawa, S., and Morita, M. (2011). Twitter Catches The Flu: Detecting Influenza Epidemics using Twitter. In *2011 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1576. Association for Computational Linguistics.

Carlson, A., Betteridge, J., Wang, R. C., Hruschka, E. R., and Mitchell, T. M. (2010). Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, page 101. ACM.

Chunxiao, W., Jingjing, L., Yire, X., Min, D., Zhaohui, W., Gaofu, Q., Xiangchun, S., Xuejun, W., Jie, W., and Taiming, L. (2007). Customizing an Information Extraction System to a New Domain. In *Regulatory Peptides*, volume 141, pages 35–43. Association for Computational Linguistics.

Cunningham, H. (2002). Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254.

Exner, P. and Nugues, P. (2012). Entity Extraction: From Unstructured Text to DBpedia RDF Triples. In *The Web of Linked Entities Workshop (WoLE 2012)*, pages 58–69. CEUR.

Finkel, J. R., Grenager, T., and Manning, C. (2007). 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.

Graham, K. and Carroll, J. (2004). Resource Description Framework (RDF): Concepts and Abstract Syntax. *W3C Recommendation*, 10(October):1—-20.

H. Cunningham D. Maynard and V. Tablan (2000). JAPE: a Java Annotation Patterns Engine (Second Edition). *Department of Computer Science, University of Sheffield*.

Harlow, C. (2015). Data Munging Tools in Preparation for RDF: Catmandu and LODRefine. *The Code4Lib Journal*, 30(30):1–30.

King, B. E. and Reinold, K. (2014). Natural language processing. *Finding the Concept, Not Just the Word*, pages 67–78.

Klusch, M., Kapahnke, P., Schulte, S., Lecue, F., and Bernstein, A. (2016). Semantic Web Service Search: A Brief Survey. *KI - Künstliche Intelligenz*, 30(2):139–147.

Murthy, D. (2015). Twitter and elections: are tweets, predictive, reactive, or a form of buzz? *Information Communication and Society*, 18(7):816–831.

Prud'Hommeaux, E., Seaborne, A., Prud, E., and Laboratories, H.-p. (2008). SPARQL Query Language for RDF. *W3C working draftd*, pages 1–95.

Republic, C. (2003). A Study on Automated Relation Labelling in Ontology Learning. *Ontology Learning from Text Methods evaluation and applications*, 123(123):1–15.

Riedel, S. and Mccallum, A. (2013). Relation Extraction with Matrix Factorization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling Relations and Their Mentions without Labeled Text BT - Machine Learning and Knowledge Discovery in Databases. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Roth, D. and Yih, W.-t. (2019). Global Inference for Entity and Relation Identification via a Linear Programming Formulation. *Introduction to Statistical Relational Learning*, pages 553–580.

Sidhu, R. and Prasanna, V. K. (2001). Fast regular expression matching using fpgas. In *The 9th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'01)*, pages 227–238. IEEE.

Takamatsu, S., Sato, I., and Nakagawa, H. (2012). Reducing Wrong Labels in Distant Supervision for Relation Extraction. In *Acl*, pages 721–729. Association for Computational Linguistics.

Wang, T., Li, Y., Bontcheva, K., Cunningham, H., and Wang, J. (2006). Automatic Extraction of Hierarchical Relations from Text. In *European Semantic Web Conference*, pages 215–229. Springer.

Zhou, L. (2007). Ontology learning: State of the art and open issues. *Information Technology and Management*, 8(3):241–252.