Literature Review: Ontology-driven Relation Extraction via an ensemble of neural network classifiers for arbitrary domains.

Kieran Bacon - 640041851

November 29, 2017

Abstract

My literature review looks at the techniques in current use to provide a method of extracting semantic relationships from text. I review these methods and descibe a new solution that aims to be more domain orientated. I provide a specification of the solution and the evaluation criteria.

We certify that all material in this document which is not our own work has been identified.

Contents

1 Relation Extraction **1**

2 Related Works **2**
2.1 Kernel based parse tree extraction . . . 2
2.2 Snowball Relation Extractor . . . 2
2.3 Semantic constraints for part-whole extraction . . . 3
2.4 Gibbs sampling to incorporate non-local information . . . 3
2.5 Exploring Various Knowledge . . . 3
2.6 Espresso: leveraging generic patterns . . . 4
2.7 Robust information extraction . . . 4
2.8 Learning to extract relations from the web using minimal supervision . . . 4
2.9 Self-supervised relation extraction from the web . . . 5
2.10 Distant Supervision for relation extraction without labeled data . . . 6

3 Literature conclusion and impact on project **7**

4 Project Specification - 2 pages **9**

5 Project Evalutation **9**

6 Conclusion **10**

1 Relation Extraction

Relation extraction is an aspect of natural language processing aimed at automatically detecting and classifying semantic relationship mentions between entities. This is achieved to that it may be expressed in a structured format from the unstructured native language, for computational purposes. Native language is very complex in its arrangement as a result of many thousands of grammatical rules and context-specific interactions. Adding to the mix, the each words possible definition and implied meaning, and there isn't any straightforward method can be used to derive a formula. Despite humanities continuous dependency on communication to function, it is a common thing for an individual to be unaware of a great deal of complexity within their own language. Regardless, there are trends within native language that govern a majority of its form, as these trends are the focus points of relation extraction. Machine learning techniques are the principal method of modelling them, and through the vast amount of textual data now available, is proving to be an effective method.

During a summer placement, I was working with a team to construct a cognitive reasoning engine capable of inferring the severity of a person's condition. Information was to be extracted from their written medical reports that originated from their hospital and GP visits. This was to accompany analytical information derived from their admission data to form structured input into the engine.

Throughout the process, we had a great understanding of the kinds of information we were looking to extract from the documents, as it was the same information experts were already looking for. Subject matter experts were tasked with constructing an ontology of relevant medical care concepts and relationships to explain that information. This formed the ground knowledge for the reasoning engine and allowed for inference to occur. The ontology only contained a subset of medical knowledge as it was all that was necessary but the ontology additionally contained aspects of the patient's lifestyle that would give rise to other conditions. The ontology didn't cover lots of concepts and relations that can be found within these documents, as they were considered irrelevant to the operation and categorisation of a patient's condition.

Many of the relation extraction methods we experimented with were disappointing as they yielded too low a precision/recall value, or could not understand the erratic English we were processing. GP reports were filled with spelling mistakes and grammatical errors and contained rather more direct and explicit notes rather than sentences. To date, I am unsure if a better solution has been found but it caught my interest as a problem to solve.

Of the methods I saw at the time, I feel that too much focus was given to general entity and relation extraction as opposed to specific relation extraction. I concluded that a solution that was formed on the basis that its output had a purpose, that there was an expectation of a secondary system, that the solution created would naturally tend to be more precise and flexible. The necessity for this perspective becomes clear when considering two separate domains like medical care and advertising. For a document detailing a child's day spent in an amusement park, the information the two domains would intend to collect would be almost entirely different. Any system should be capable of yielding two entirely different collections of 'relevant' information as the domain and requirements change. With this in mind, I believe that relation extraction shouldn't be general. In practice, I believe that information about a particular domain is known to a user before a form of information extraction is implemented. This information can be expressed rather efficiently within an ontology, and should form the constraints of any relation extraction system.

I arrived at the conclusion that a supervised method of relation extraction, based on an ontology definition of a domain, would better solve the problem while being more adaptive and useable. A system like this would be able to better utilise the classification power of current machine learning techniques, focusing learning on relevant information. Furthermore, the method would be suitable to model domains that could be considered to be simplistic or highly specific without any issues, as training would be accurate and representative of each domain.

I shall be examining the relevant literature on techniques currently in use to solve relation extraction and use them as inspiration for a new ontology-based method. I will present the specification for my proposed project and provide the evaluation criteria by which it shall be determined a success. Additionally, I will indicate a series of investigations I wish to conduct to measure its flexibility and versatility as a method.

2 Related Works

A relation extraction application aims to have: *good performance*, high precision and high recall with a reasonable time constraint; be useable with *minimal supervision*, limiting the need for human interaction; have *breadth*, to be applicable in a variety of situations; and have *generality*, to be applicable to many different kinds of relations in different domains.

2.1 Kernel based parse tree extraction

Many variations of parse tree implementations have been made to help solve relation extraction. Among them, feature-based methods achieve certain success by employing a large amount of diverse linguistic features, varying from lexical knowledge, entity related information to syntactic parse trees, dependency trees and semantic information. [9] [8] However, they have found it difficult to effectively capture structured parse tree information and for further performance improvement, this challenge needs to be overcome. [8]

G. Zhou et al (2007) presnts an alternative method to feature-based parse tree implementations that achieved comparable performance with other 'state of the art' SVM implementations with linear kernels. The method aims to resolve two main issues that have bothered other parse tree implementations while introducing a novel way of comparing similarities between features trees.

A primary issue for other tree implementations is that they look specifically at the information between the entities to classify. For relationships like 'marriedTo', we can expect to get statements like "John and Sarah have gotten married" which explicitly defines the relation. Conventional tree parsing would not be capable of defining this relationship as the necessary context falls outside the tree. A follow-on issue is that relations that have a similar structure (little content between the entities) prove hard to manipulate, as their features tend to correlate highly with most relations.

Their adaptation is to use a context-sensitive convolution tree kernel that includes the parsing of nearby context words into the tree. G. Zhou et al (2007) have devised a new algorithm to include ancestral nodes as a possible route of computation when evaluating a parse tree to make use of this information. Their specilised convolution kernel counts the number of common sub-trees (substructures) as the syntactic structure similarity between two parse trees and is used as the metic to perform analysis.

In conclusion, it was though that the parent and grandparent nodes of a sub-tree contains the majoirty of the information necessary for relation extraction. Considering more ancestral nodes did not yield improvement, a claim supported by G. Zhou et al (2005). For the various testing implementations, they were able to achieve general precision scores of 77% with recall of around 63%.

2.2 Snowball Relation Extractor

E. Agichtein (2000) presents a semi-supervised method of relation extraction named the "Dual Iterative Pattern Expansion". The methods aim was to extract a structured table of relations from a collection of documents similar to those found on the web. Inevertably the method works best for evaluating these data sources as the table tuples it hopes to find appear in uniform contexts. DIPRE exploits this redundancy and inherent structure in the collection to extract the target relation with minimal training from a user.

The method generates a collection of extraction patterns through a process of recombination and generalisation of patterns over iterations of learnings. The initial seed instances are provided which in turn gives it the semi prefix. These instances are not patterns themselves, instead they are positive entity pairings for a relationship. e.g. "*HQ(Microsoft, Redmond)*". These pairings are used to search the text to collect context information for any segment that contains them. The information is collected into tuples of information describing the sequences of words to the left, right and between the entities. Tuples that have similar or equal length middle sequences undergo combination and a pattern is formed.

These patterns are then scored by comparing the number of positive relations it extracts that were given in the input, with the number of relations it returns. A selection of the highest scoring patterns and then used to find new segments of the document to forms the seed tuples for the next iteration.

Snowball was evaluated on a collection of 300,000 newspaper articles and proved to be able to produce high-quality tables for various selections of the corpra. There is evidence to suggest that due to the scalable evaluation methodology that these results will be consistent on other collection. The method seemed to have a trade off between precision and recall however an optimum was found that yield around 80% for both.

2.3 Semantic constraints for part-whole extraction

Meronomy is a hierarchy that describes the relationships formed between entities that indicate that one entity is part of another entity. One of many entities that form a whole. e.g. 'handle' is a part of a 'door'. Girju et al. (2003) have devised a supervised method for identifying this information within the text by constructing a decision tree filled with if-then rules. Each rule is learnt through a process of searching a hypothesis space with varying complexity of assumption. When a hypothesis is found that is relatively consistent with the data, it is introduced into the tree structure. Rules that yield the highest information gain are places closest to the root to reduce wasted computational time on incorrect hypothesises.

Method input is a set of positive and negative meronomy examples and targets. The lexical patterns of the training examples and the semantic relations detected by selectional constraints form a starting point for the process.

It is known that the relationship has two types of lexicosyntactic patterns: Explicit constructions where the sentence literally describes the membership of one entity to another "*The door is made of steel*"; and implicit construction, during identification of the entity, you identify its membership to another "*door knob*". Implicit construction is very often confused with ownership however and for this reason, context bases patterns are necessary to identify them. As a product of this innate trend of the relation instances, two types of pattern emerge to extract these relations: Phrase-level where the concepts within a phrase or paragraph; Sentence-level where the relation is intrasentential.

Girju et al. (2003) have been able to achieve precision and recall rates of 84% and 98% respectively which is exceptional. However as this method takes advantage of the innate properties of this specific relation its use is limited. I envision this method being extremely useful in the identification of the entities or concepts withing a named entity recogniser, or ontology builder.

2.4 Gibbs sampling to incorporate non-local information

Finkel et al. (2005) identify that most statistical models that perform natural language processing tasks tend to represent only local structure. They found the constraint to be paramount in allowing the problem to be tractable but felt it was a key limitation generally. They believed that relaxing the requirement of exact inference and instead utilising approximate inference algorithms would allow for a tractable non-localised model to perform classification.

Finkel et al. (2005) set about constructed a method to perform approximate inference in factored probabilistic models to statistical predict the entity classification of tokens within a text. The method makes use of the Monte Carlo method referred to as 'Gibbs sampling' to produce Markov chains across the corpus. The nodes are in spaces of possible variable assignments such that the stationary distribution of the Markov chain is the joint distribution of the variables. The transition probability of the Markov chain is the conditional distribution of the label at the position given the rest of the sequence. This quantity is easy to compute in any Markov sequence model but the entire process is rather inefficient.

As a method, Finkel et al. (2005)'s would likely perform well as a co-referencing method for relation identification that does not contain two entities. The method was able to achieve averagely 89% precision on the the CoNLL named entity recognition task, and the CMU Seminar Announcements information extraction task.

2.5 Exploring Various Knowledge

G. Zhou et al (2005) has conducted an investigation into the merits of employing diverse lexical, syntactic and semantic knowledge in feature-based support vector machine implementations. Support vectors were chosen to perform relation extraction as Zhou felt that they offered the best comparative learning aspects of machine learning techniques.

As part of a preprocessing step, pronominal mentions are replaced with the most recent non-pronominal entity mention for a basic implementation of co-referencing. Semantic information from various resources, such as WordNet, is used to classify important words into different semantic lists according to their indicating relationships.

G. Zhou et al (2005) demonstrated that the base phrase chunking feature information contributes more to performance than any other syntactic functions. Full parsing of the context information is shown to offer next to no performance enhancement, as relevant information can be found in shallow dives. They suggested that this could be a product of the ACE corpus that their investigation was evaluated on.

G. Zhou et al (2005) have designed their system to maintain a classifier for every relation imposing a 'one vs. all' strategy to separate the classification of relations. This method has shown the be competitive

in performance to the classical $\frac{k}{2}(k-1)$ classifer arrangement.

Through a combination of the feature vectors produced within the classifer, G. Zhou et al (2005) were able to show improvements of previous feature based systems and tree kernel methods. Precision was marginally better with a score of 84.8% but benefited from a greatly improved recall rate of 66.7%.

2.6 Espresso: leveraging generic patterns

P. Pantel et al (2006) theorised that given a set of patterns that can extract target relations with high precision but low recall. Any generic pattern can be scored and ranked by comparing the instances they return with the instances identified with the precise set. As a result, P. Pantel et al (2006) constructed a method that performs well demonstrating high precision and recall values across the ACE corpus.

The method is a minimally supervised boot-strapping algorithm that takes as input a few seed instances of a particular relation and iteratively learns surface patterns to extract more instances. These seed instances are used to generate the high precision patterns that are used throughout by using an arbitrary pattern learner. These patterns are used to find segments of text that shall form the bases of the next relation patterns. Segments have their terminological expressions replaced and their contents generalised before being used to evaluate the locus. The method maintains a collection of generic patterns with each iteration that are shown align with the initial precise set, they are described as reliable generic patterns.

A consideration for P. Pantel et al (2006) was the difficulties associated with operating on small corpora. As a solution, instances undergo syntactic expansion to form multiple initial patterns. Entities and terminological expressions are replaced by appropriately verified sources on the web.

2.7 Robust information extraction

M. Surdeanu et al (2007) have developed a classifier that used online learning techniques to generate a large-margin perceptron algorithm to separate relations instances. With an enthesis on simplicity and robustness, M. Surdeanu et al (2007) have used only NLP preprocessing tools that have been proven to work well on any corpus size, such as part of speech tagging and chunking. What they investigated was the practicality of using variants of the perceptron algorithm for all learning tasks of a relation detection application. From this, they designed a novel architecture that is effective and efficient while mitigating errors that arise in early stages of classification.

Training points have their part of speech tags and chunking information extracted before heading into an entity mention detector. The detector implements a sequence tagger and attaches marks to the tokens to describe positive entity instances. At this stage, if the detector is unsure about the classification of any entity, multiple instances are created and allowed to permutate through the system. Instances then travel into the relation mention detection system to train the system as to their trend. It is expected to receive very unbalanced data, receiving majority positive examples. When conducting an evaluation, every possible relationship that can be expressed within a passage is generated, a single consistent solution is formed later by inference. During the inference stage ambiguities within the entity-tagger are resurfaced to add additional validity to a particular relations confidence.

Unexpectedly, perceptron algorithm responsible for relation classification learns not only when a prediction is incorrect, but also when the classifier has insufficient confidence in the prediction. Uneven margins separate the feature space, relations are then predicted positive or negative if they fall outside the classifier's margins.

The method was tested on the ACE 2007 English corpus. The method was noted to be exceptionally fast, collectively taking around an hour to complete training. Overall, the perceptron algorithm was able to achieve a precision score 20% better than the linear kernel-based SVM implementation, taking 5 minutes the SVM's 15 hours. M. Surdeanu et al (2007) method did suffer in recall however but consistently produced better F1 scores.

2.8 Learning to extract relations from the web using minimal supervision

R.Bunescu et al (2007) have developed a method for relation extraction by extending a weaker form of multiple instance learning using support vector machines and string kernels. The method aims to construct a feature space whose dimensions relate to the sequence of words within an extracted relation instance. The instances are raised to a high dimensional hyperspace and classified linearly. The intention is that the system learns the trends that make a mention of the entity pairings positive or negative and therefore can be used to classify any given document with the generalised information.

The input is solely entity pairs for target relations. These pairs either describe a positive or negative instance of the relation. Sentences that contain a relation pairing are recorded from the internet to form the training data of the system. No pattern is used to find sentences, simply a match on the entity is used, therefore there is no initial confidence in any given data point. Sentences that do not contain pairs that may express a relationship are removed. The sentences that are recorded are split into positive and negative bags depending on their pairing orientation. The corpus is assumed to be sufficiently large and diverse such that it is highly likely that the positive bag contains at least one sentence that explicitly asserts its relationship. The negative bag is assumed to be populated primarily with sentences that do express the negative relationship, as any mention would be an indication of the relationship not being followed.

R.Bunescu et al (2007) have deliberately avoided the use of syntactic analysis as it has often given them unreliable results. They demonstrated that for a variety of documents on the internet, such processing tends not to yield a benefit.

These sentences are then used to train the Support vector machine, sentences are broken down into a static number of tokens along with the entities. Tokens are chosen by a ranking system, stop words and punctuation are removed to form expressions of the original instance, these can be learnt by the SVM. This method introduces two forms of biases: High correlation between a subset of tokens and a relation's entities; and high correlation between a subset of tokens and the relation itself. In doing so, sentences representing a negative instance that contain these tokens can be mislabelled, therefore a greater enthesis on regularisation is required.

The method was motived by the high success rate of other SVM implementations for relation extraction. This method can be transformed into a standard supervised method by labelling all the contents of the bag before processing.

The evaluation was done on two relationships with 20 entity pairs when aggregated. Training data was collected by searching the internet attempting to match any sentence that contained a pair with a maximum of any seven words between them. Searches were limited to English written documents.

2.9 Self-supervised relation extraction from the web

SRES classifies relations from self-taught extraction patterns that are derived from unlabeled text documents. The patterns are formed by generalising the instances of relations extracted by prior patterns by identifying and extracting general keywords and attributes in those sentences. The greatest scoring patterns are kept, similar to an evolutionary algorithm implementation. Initial patterns are required to start this process and they constitute the entire input of the system. They are used to generate pattern seeds which are iteratively fed into the pattern learning aspect of the system. The pattern leaner is simplistic in nature as to reduce undue overfitting. These patterns are short descriptions of the relation and its attributes, they are simplistic to reduce required human involvement.

A relation description is comprised of: a collection of relation scheme indicating how it might be found in the text; key tokens that might be found within those representations; and behavioural indicators that give insight into the nature of the relation. From this, the patterns are used to extract valid sentences from the web (or provided documents). Valid sentences along with their patterns are used to help generate new seeds that better represent the relations. Large collections of patterns are formed by the cross combinations of the sentences. Patterns that are either redundant or overly specific are removed before the process repeats.

These patterns are used in a simple substring search of the text to extract sentences of interest. Patterns regularly contain gaps within them that may match with any sequence of tokens, this general matcher allows for large amounts of data to be collected. It is an assumption within the method that there is only limited redundancy within the corpora. The initial patterns extract a small training set of relation instances with high confidence. These sentences are broken down by the NER element of the system into tokens and keywords which have the potential to become pattern factors.

Negative instances of relations are generated by altering positive instance attributes with 'acceptable' replacements from their original sentence. Keywords and arguments are edited with few limitation. Mislabeling is thought to be a small occurring during this process.

Patterns are scored via a monotonic function formed from the positive and negative instances of that are generated from their operation. The number of positive instances over the number of negative instances squared, to be exact.

In general, the process has bad scaling ability due to the expensive generation and combination of extracted sentences to form new potential patterns. A knock on effect is the evaluation of many

redundant patterns is conducted which is a further strain despite being linear in operation. Only five relation types were examined in the evaluation stage. The process by which negative instances are formed equally introduced issue as sentences that contain two positive instances can end up being added to both the positive and negative class, furthermore incorrectly positive patterns may suggest positive instances as negative instances through this process.

2.10 Distant Supervision for relation extraction without labeled data

Identifies [10] that supervised relation extraction suffers from: Overhead due to gaining labelled data; Classifiers tend to be biased towards the text domain.

Unsupervised method is to perform information extraction, perform clustering and simplify the strings of free text to produce relation strings. Sufferes from mapping extracted relations to particular relationships.

Boot strap learning. Using a smaller seed instanct pattern, new pattern instances are generated in a iterative process where the pattern population is continually applied.

Suffers from low precision and semantic drift.

An extention to the paradigm by X for exploiting WordNet to extract hypernym (is-a) relations between entities. Similar to the use of w3eakly labelled data in bioinformatics

Alogrithm uses freeebase - large semantic database of concept and relationship pairings. Essentially a gigantic ontology. The primary intuition is that concepts that appear within sentences are likely to express these relationships. From this lots of relationships can be generated (most of which are noisy), these are then combined in a logistic regression classifier (the logistic regression returns category assignements like a classifier would)

Relationships between entities is supported within this method by aggregating the features of an entity pair from throughout the literature. This is meant to decrease uncertainty on their generated label.

Syntactic features are known to improve perforamce of supervised information extraction. (when using labeled ACE data)

Method: All entities are identified usaing a named entity tagger. If a sentence contains two entities, and if those entities relate to any of the relationships expressed in the Freebase lexicon, the features of the sentence are added to that relations feature vector.

A multiclass logistic regression classifier is used to learn weights for each noisy feature, which hopes to conpensate for incorrect relation identification. Identical relations from different sentences are combined to aggregated to form a 'richer' feature vector.

Within testing, the same entity tagger is used to find concepts. Features from the sentences where entity pairs exists are collected. This is run through feature extraction and the regression classifier is used to predict the relation name.

Lexicon Features are of the form:

- sequence of words between the two entities + part of speech tags
- entity order
- k word window before the first entity + part of speech tags
- k word window after the second entity + part of speech tags

all these components form the lexical feature.

part of speech tags are assigned by a maximim entropy tagger, that was trained on the Penn Tree bank. Omitting all non verb words and all function words, did not yield enough precision improvement to justify the computational expense.

Syntactic features:

- A dependancy path.
- A window for each entity not part of the dependancy path.

A **dependancy parse** consists of a set of words and chunks that are linked by directional dependancies.

Named entity tag features. They use the stanford four class named entity tagger

Uses a set of generic extraction patterns and automatically[10] instantiates rules by combinfing those patterns with user supplied relation labels. These patterns provide a structure and along with possible

expectations of the relation, and space for matches to occur. KnowItAll then passes these through a mechinism that controls the quantum of search, and merges redundant extractions while assigning probability to each extraction bases on frequency of extraction.

This method tends to have low recall due to a wide variety of contects describing a relation. KnowItAll also has a simple pattern learning scheme that builds on the generic extraction mechanism and its why it is refered to as a bootstrap method.

This process creates a set of positive training sentences by downloading those that contain both argument values of a seed tuple and the relation label. Negative training is created by downloading sentences with onlu one of the seed argument values and considering a nearby NP as the other value. This does not guarentee that the negative examples will be false. Works well in practise.

The induction process tabulates each occurance of tokens surrounding the arugment values. The k tokens to the left of the first instance, the tokens between the two arguments and k tokens to the right of the second argument.

3 Literature conclusion and impact on project

If an assumption is made that the user of an information extraction method intends to use the output as a form of input for a secondary system. The relational information is to be used to perform inference, or evaluate some form of analytical function. Then it stands to reason that a method of relation extraction is supervised, as it is not the entities and the relations that are being learnt. What is consistent is that each user has a particular domain of use, and they have a clear understanding of the information they are aiming to extract from their text. Textual documents typically contain large amounts of redundant information, and any information that does not provide any value to the secondary system should be ignored.

The aim is to understand what relations out of the relevant, do actually occur in a document, and to what frequency and with what confidence. A supervised method would come with a requirement for a large, annotated corpora. Producing such a corpora is considered to be rather challenging and time-consuming and has been the main aspect semi-supervised methods have tried to avoid. [12][7][11] Such methods have instead decided to limit user input to a collection of seed instances, or tuples of positive and negative instances of single relations.[1] [12][5] A by-product of such a reduced input is that the methods look to extract additional entities and relation definitions during there operation and inevitably introduces false information. These systems are aware of this and focus attention on ensuring the method is flexible enough to accommodate.

I believe that this is a negative aspect of these systems as it does not align with our original assumption. For a domain where I know all the entities and relationship I want to identify, iteratively expanding the scope of the domain is counter-intuitive. These models exclaim that irrelevant relations can be filters out at a later point for specific tasks when classification has been conducted, which is rather computationally inefficient. As these methods aim to reduce the requirement for user input, I find it intriguing that not more enthesis has been put in structuring the seed information in a way that helps to include as much information as possible. I intend to use an ontology to encapsulate the information about the domain. In doing so I ensure that the system does not expand the scope of the entities or relations while I maintain large quantities of information within a fraction of the space. If we imagine that we have a basic ontology like:

```
Entities:
    "Person": {}
    "Paul": {"parents":["Person"]}
    "Sarah": {"parents:["Person"]}
    "John": {"parents":["Person"]}
    "Dave": {"parents:["Person"]}
    "Physical":{"readable":False}
    "Vehicle":{"parent":["Physical"]}
    "Monster":{}
    "Car":{parents:["Vehicle"]}
    "Monster Truck":{"parents":["Vehicle","Monster"]}

Relations:
    "encourages": {"domain":["Person"],"target":["Person"]}
```

```
"pushes": {"domain":["Person"],"target":["Person","Car"],"self":False}
"loves": {"domain":["Paul", "Sarah"],"target":["Paul","Sarah"], "self":False}
"admir": {"domain":["Paul"],"target":["Sarah","Car"]}
```

Each of the lines can encode multiple pieces of important information in what feels like a natural structure. Entities of relevance are listed in a simple manner and could be easily generated for a task. They would reference their parents so that relations may cascade down onto them without the need for explicit instruction. They can also indicate if they are expected to be readable within the text. For different domains, entities may be present in the structure but are not intended to be read or interpreted by the classifier. This may be because they don't yield relevant information. Allowing this functionality goes a long way in reducing the complexity of the required classifier.

The 'encourages' relationship indicates that between any combination of Person and its sub-entities, one can "encourage" the other. This single expression expands to 25 relationships in total, each of which can be read in the text. As each domain entity can link with each target entity we should also be extracting information like 'Paul encourages Paul' which is a desired property for my situational domain.

I am making the assumption that the entity 'Person' has been left readable. We would expect to read information such as "Paul encourages a person to ..." in our document. If that were not the case, we would only expect 16 relations to derive from the first relation.

Relation 2 makes use of the additional 'self' tag that indicates that for the expansion of the relation expression, relation instances that link an entity to itself are not valid. This would ensure the 5 relations of a person pushing themselves were not generated or looked for. Relation 2 also indicates that there is an asymmetrical relation between Person and its sub-entities, and the entity Car.

Relation 3 demonstrates the ontologies ability to contain high and low-level definitions in conjunction. A symmetric relation has been formed between two sub-entities of person. So despite being entity siblings, John and Dave do not participate in any 'loves' relationships. Additionally, as the 'self' tag has been applied, Paul nor Sarah love themselves (shame).

Relation 4 simply explains an asymmetric relationship, Paul is able to admire both Sarah and the car but is not admired in return. B. Rozenfeld and R. Feldman (2008) used keywords to allow for expansion to occur on their initial instances in a similar manner. The ontology by definition would contain this type of information natively, allowing for greater input with little effort.

Various implementations of Support Vector Machines were the prefered choice for information and relation extraction in the reviewed documents.[1][6][11][3][8] Performance is generally derived from the kernel of choice and since many have had relatively good success, focus has been on the adapting previous works to include additional lexical and semantic knowledge into the classification.[8] However, there is evidence to suggest that the inclusion of such information yields little improvement over the simple chunking and shallow dive techniques. [8] M. Surdeanu et al (2007) has a much more promising technique that has shown to produce on par or better than the common SVM approaches in a greatly reduced time. They make use of only the part-of-speech tags and chunking information to train perceptron based large margin classifier while making use of simple and robust NLP systems. Additionally, they allow multiple hypotheses of a relation to flow through the system to be inferred at a later point when all information can be aggregated.

With the user's considerations in mind, a combination of M. Surdeanu et al (2007)'s and G. Zhou et al (2005)'s techniques to produce a simplistic but powerful classifier resonates with me best. Implementing the '*one vs. others*' strategy with neural network based classifiers using simplistic and robust NLP methods is expected to be very promising.[12] Generating an ensemble of neural network classifiers that work specifically to represent a relation group (a single relation with multiple domains and ranges) shall allow for tailored classification. As a result, for statements that can contain only a single relation, only a single classifier has to be used to determine its presence. For a statement that might contain multiple relations between the same entities, a classifier for each relation can be used to suggest a classification, the class with the highest confidence can be chosen. Another benefit that presents itself from the ensemble approach would be the ability to introduce new relations into the classifier without affecting availability. If the ontology changes to include additional information, classifiers within the system are not affected, a new classifier for a new relationship can be produced easily. [13]

I do not intend to use a Named entity recogniser during the operation of the classifier as that information is expected to be included in the ontology. Instead, I intend to record a locus of entity patterns

that are collected from the annotated documents. For each entity, a dictionary of phrases will be produced to identify them, and finding segments of a document that contains a relation will be as simple as matching words with entities through those patterns. For this method, I would expect a very high recall rate, although it falls prey to incomplete annotated documents. Relations would not be identified if the entity is described in a way not labelled in the annotated documents (misspelt entities as an example). To counter this, a method of co-referencing the annotated documents would also need to be generated. Finkel et al. (2005) suggests a way of finding entities probabilistically by inference and this could be used as a to resolve entity inconsistencies and co-referencing problems.

Many techniques spend time ensuring that highly tokens are removed or penalised within the system as to ensure that overfitting does not occur. I will be investigating the impact of such techniques on the output. My intuition tells me that for the multiple classifiers approach will not be hindered. When this method has been used in literature, it was usually accompanying a single classifier arrangement. Additionally, I aim to investigate the effect of spelling and grammatical mistakes within both the training and test sets of documents. The aim is to be as flexible and useful to local and global domains.

4 Project Specification - 2 pages

Structure The project shall be a python package that enables the creation of the project's relation classifier. The classifier a simplistic API primarily offering functionality to train and predict on annotated sets. A classifier will take as part of its constructor the ontology that describes the domain. For training, a set of annotated document objects will be taken along with their targets. The inputs to the object shall be in the form of an ontology object/file and a collection of annotated sentences/paragraphs.

Ontology The ontology provides information about the entities and relations contained within the domain. Additionally, it indicates a hierarchy of entities to allow for cascading of relations. e.g. *Person drives vehicle* can be cascaded to *Paul drives BMW*. This helps ensure the document is relatively small and simplistic, facilitating its development. It provides a powerful tool for the expansion of relations over the domain in a deterministic manner without losing information while being as general or specific as the user intends.

Technique Classification is to be performed by an ensemble of multilayer perceptron algorithms, similar to those expressed in M. Surdeanu et al (2007). Each classifier shall be tasked with determining the confidence of a particular relationship expressed within the ontology. They shall take as inputs tokens of a sentence that will be anchored around the entities. The number of inputs of a classifier shall be determined by K and shall express the number of context words around the pair to include. An input shall be the words before the first entity, the words between the pair of entities, and the words following the second entity. The entities themselves do not need to be included as are assumed by the classifier. The classifier can then operate on any combination of entities within the ontology that inherit from the entities of the relationship, allowing one classifier to correctly evaluate many entity pairings.

Training Training the system focus on paragraphs/sentences that contain entities that can express relationships found within the ontology. For entity pairs, the relations are identified and the classifiers responsible for representing them are found. Words within a sentence shall be tokenized, and their features shall include Part of speech tags and

Prediction The system is to return a collection of relations found along with the segments of text they originated from, their index in the text, and their confidence. During the evaluation section, the input document will be subjected to various refinement processes designed to minimise error. Spelling will be corrected, co-referencing shall be conducted and entities inserted.

5 Project Evalutation

Recall The recall rate is the number of relevant relations that have been identified over the number of relations present within a text. Its a measure of a classifiers ability to extract sections

of the text that contain a relation. It ignores the classifiers wrongful attempt to classify irrelevant relations (something my method attempts to minimise). I aim to have a recall better than that produced by M. Surdeanu et al (2007)'s implementation of 63.4% to show that my ensemble is an improvement on the perceptron inspired large margin approach. However, as a supervised task, I would expect it to be on par if not better than Tree Kernel based who achieved 65.6%. [14]

Precision The number of correctly classified relations over the number of extracted relations. How accurate are the classes predicted by the classifier. To be consistent with the evaluation methods of other methods reviewed, I will be evaluating the rounded confidence of the classifer. Supervised methods have been able to peak at around 79%, where as Unsupervised methods around 67%. I would be aiming to produce competitive supervised results looking to produce results between 70% - 80%

Hospitable I would like to compare the effect on precision and recall when conducting relation extraction of a range of domains. classifiers that make use of WordNet and Freebase systems to identify entities and relations are likely to not have the ability to adapt. I would like my classifier to remain relatively consistent with its scores.

ACE ACE is a corpus of annotated information and relation extraction texts used as the standard for comparing NLP techniques. I will be constructing an ontology to adhere to their annotated documents and producing my precision and recall values for comparison.[4][2]

6 Conclusion

Relation extraction has the potential to become an integral part of a variety of systems. The ability to understand the semantic information in a collection of text is a requirement for inference to occur on its contents, and systems that aim to be user-friendly (question and answer bots) require this capacity. However, I feel that most work that has been conducted on the subject has been focusing their efforts on extracting every possible relation rather than the relevant. If we assume that language has patterns that allow general relations to be extracted, it should follow that there are trends in the subsetted lexicon of a domain. Utilising a system that is focusing on domain-specific relations should yield better and more useful results.

Controversially, I prefer recall as a metric for success over precision due to its utility in a business scenario. Taking my placement as an example, lots of time-consuming work goes into the processing of medical documents that can he hundred of pages long. If we imagine a system with low precision but high recall. The shortfall of low precision can be resolved by setting a high threshold of confidence before asserting a relation. The end result is that there is a high amount of documents that are required to be reviewed in their original manner. However, having recall sufficiently high, a user can be confident that the system has identified all the relevant information of the document. A summary can be drafted, and data can be labelled for future progression of the model. For the alternative, the high precision does not make up for the lack of information being extracted from the document. Missing some critical information could cause untold financial and personal turmoil. In this case, the entire document would still need to be reviewed, no documents could be vouched for, and it is not an issue that can be improved as dramatically with additional data points. Case and point, the precision metric is derived from the number of correct classifications given the relations extracted, for a recall of 0% we would obtain a precision of 100% arbitrarily.

References

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM, 2000.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

[3] R. Bunescu and R. Mooney. Learning to extract relations from the web using minimal supervision. In *ACL*, pages 614–621, 2007.

[4] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. Strassel, and R. M. Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *LREC*, volume 2, pages 837–840, 2004.

[5] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[6] J. R. Finkel, T. Grenager, and C. Manning. 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, 2005.

[7] R. Girju, A. Badulescu, and D. Moldovan. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 1–8. Association for Computational Linguistics, 2003.

[8] Z. GuoDong, S. Jian, Z. Jie, and Z. Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics, 2005.

[9] N. Kambhatla. 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*, page 22. Association for Computational Linguistics, 2004.

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

[11] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics, 2006.

[12] B. Rozenfeld and R. Feldman. Self-supervised relation extraction from the web. *Knowledge and Information Systems*, 17(1):17–33, 2008.

[13] M. Surdeanu and M. Ciaramita. Robust information extraction with perceptrons. In *Proceedings of the NIST 2007 Automatic Content Extraction Workshop (ACE07)*, 2007.

[14] G. Zhou, M. Zhang, D. Ji, and Q. Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.