Extracting Meta Statements from the Blogosphere

Filipe Mesquita
Department of Computing Science,
University of Alberta, Canada
mesquita@cs.ualberta.ca

Denilson Barbosa
Department of Computing Science,
University of Alberta, Canada
denilson@cs.ualberta.ca

Abstract

Information extraction systems have been recently proposed for organizing and exploring content in large online text corpora as *information networks*. In such networks, the nodes are named entities (e.g., people, organizations) while the edges correspond to *statements* indicating relations among such entities. To date, such systems extract rather primitive networks, capturing only those relations which are expressed by direct statements. In many applications, it is useful to also extract more subtle relations which are often expressed as *meta statements* in the text. These can, for instance provide the *context* for a statement (e.g., "Google acquired YouTube **on** October 2006"), or repercussion about a statement (e.g., "The US **condemned** Russia's invasion of Georgia"). In this work, we report on a system for extracting relations expressed in both direct statements as well as in meta statements. We propose a method based on Conditional Random Fields that explores syntactic features to extract both kinds of statements seamlessly. We follow the Open Information Extraction paradigm, where a classifier is trained to recognize any type of relation instead of specific ones. Finally, our results show substantial improvements over a state-of-the-art information extraction system, both in terms of accuracy and, especially, recall.

1 Introduction

Current information extraction systems expose the content of large text corpora as *information networks* where nodes are named entities (e.g., people, organizations) and edges represent relations among such entities. Such information networks are powerful metaphors for visualizing large complex systems, such discussions and comments made collectively by users in a shared social media space. In recent work, (Mesquita, Merhav, and Barbosa 2010), we explored the use of these ideas towards unveiling interesting conversations in this space, obtaining an information network built upon 25 million blog posts from the ICWSM Spinn3r dataset (Burton, Java, and Soboroff 2009). This network contains entities and relations frequently cited in the blogosphere between August and September of 2008. Our experimental analysis indicated accuracy results comparable to the state-of-the-art applied to curated corpora. As an example, Figure 1(a) illustrates an information network about the conflict between Russia and Georgia, a popular topic among bloggers at that time.

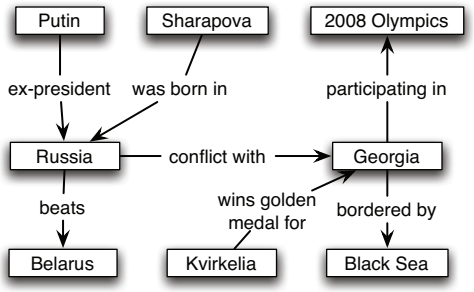
Despite their effectiveness, information networks are somewhat primitive, as they are unable to represent certain interesting, more subtle relations expressed in the text. For example, they cannot naturally express that The Associated Press (AP) **reported on** the conflict between Russia and Georgia, since the conflict is represented by an edge. Thus, the relation between AP (which is a node in the network) and the conflict (which isn't) cannot be directly represented.

Using the terminology of knowledge representation, each edge in an information network can be viewed as a *statement* about the entities it connects (W3C 2010). In this terminology, the relation between AP and the conflict can be represented through the recourse of a *meta statement*: defining a statement (AP reporting on) about *another* statement (the conflict). In knowledge representation terms, this is also called *reification* (Yang and Kifer 2003).

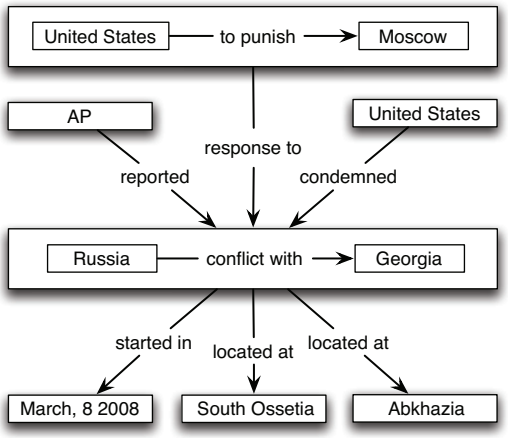
Figure 1(b) illustrates a *reified network* presenting both the statements and meta statements about the conflict in Georgia. Observe that, besides stating that AP reported the conflict, this network also captures: (a) the conflict's repercussion in the United States, (b) its potential consequences (the threat of retaliation from the United States) and (c) some context for the conflict (i.e., date and place). We posit that such networks can provide even richer (and hence more useful) information networks. In this work, we report on a system aimed at extracting reified networks from the blogosphere. This includes extracting statements and meta statements from natural language text found in blog posts.

Problem statement and Challenges. The problem addressed in this work is that of accurately extracting reified information networks from natural language text in social media websites (i.e., the blogosphere).

In such an environment, extracting statements and meta statements from text presents many challenges. First, recognizing and disambiguating entities from a large collection of documents is a difficult task on its own (Jurafsky and Martin 2008). Second, achieving high quality extractions is very difficult, given the complexity of the English language (Sarawagi 2008) and the diversity of writing styles in the blogosphere. In particular, even with well-written text, as illustrated in detail later in the paper, the nested structure



(a) A standard information network exposing facts around the representation of the conflict between Russia and Georgia (entities), mentioned in the ICWSM Spinn3r dataset.



(b) A reified version of the same network, showing meta statements about the conflict in Georgia.

Figure 1: Examples of how statements are expressed in information networks. Our reified network extend the traditional one by allowing statements about other statements.

of meta statements brings problems not found in traditional, "flat" relation extraction. For instance, one must determine whether a relation expressed in a sentence concerns an entity or another statement, expressed in the same sentence, containing such entity. This ambiguity undermines a classifier's ability to differentiate between direct and meta statements. Finally, combining different features (e.g., words, part-of-speech tags and parse tree) in order to achieve meaningful results is a nontrivial exercise in modelling (Sarawagi 2008).

In our work, we follow the seminal approach in TextRunner (Banko and Etzioni 2008), a state-of-the-art Open Information Extraction (OIE) system. Namely, we rely on a supervised method for handling the actual text—we use Conditional Random Fields (CRF), and exploit syntactic features found in parse and dependency trees obtained for the sentences. It should be noted that our goal here is to extend the use of CRF as described by Banko and Etzioni, in order to handle both statements and meta statements. If successful, our model could be used within the larger framework in TextRunner, which encompasses both the self-supervision scheme for training the CRF, as well as the post-processing module that further checks the plausibility of the extracted facts.

One interesting application of our information networks is competitive intelligence. Competitive intelligence is a business practice that includes collecting and analyzing information about products, customers and competitors of an industry. The blogosphere is one of the most valuable information sources for this type of analysis; however, it is infeasible for a human to read every relevant blog post. Yet, one is expected to take business decisions based on all information available. Our networks provide a solution for this information overload problem by allowing one to analyze and visualize the most cited entities and relations in the blogosphere.

Contributions. In this paper, we propose a relation extraction method that applies Conditional Random Fields to extract direct and meta statements seamlessly. We show that the original CRF model in TextRunner, O-CRF, lacks discerning power to accurately handle both kinds of statements, and provide an explanation of why this is the case. We show the need for relying on more sophisticated syntactic features, and propose a new CRF model, meta-CRF, that is powerful enough to help the classifier differentiate between direct and meta statements.

In quantitative terms, our experimental validation of meta-CRF on a sample of the ICWSM Spinn3r dataset, shows substantial improvements over O-CRF (when both models are trained with the exact same training examples). More precisely, meta-CRF outperforms O-CRF considerably in terms of recall, and substantially in terms of accuracy (over 20%). On the other hand, a small loss (3%) is observed in terms of avoiding false-negatives.

2 Related Work

This section discussed related work from two fields: information extraction and knowledge representation.

Information Extraction. Relation extraction is an important problem in information extraction that has attracted much attention recently. Some studies consider extracting relations with any number of arguments (McDonald et al. 2005; Wick, Culotta, and McCallum 2006; Xu, Uszkoreit, and Li 2007). However, most approaches consider the problem of extracting relations between two arguments. This problem is traditionally defined as classification problem (Bollegala, Matsuo, and Ishizuka 2010): given a relation R and a pair of entities in a sentence S, does S asserts R between this pair of entities? Supervised systems use manually labeled examples to train a classifier for each relation. This classifier is either based on extracted features (GuoDong et al. 2005) or kernel functions (Zelenko, Aone, and Richardella 2003; Culotta and Sorensen 2004; Bunescu and Mooney 2005). Bootstrapping systems require significantly less training data. These systems discover new relation instances by using a small set of entity pairs (Brin 1998; Agichtein and Gravano 2000) or hand-crafted extraction patterns (Etzioni et al. 2004). A limitation of these approaches is that they scales linearly with the number of relations.

Our approach is based on Open Information Extraction (OIE), the paradigm of extracting *unanticipated* relations (Banko and Etzioni 2008). OIE systems (Banko and Etzioni 2008; Zhu et al. 2009; Hasegawa, Sekine, and Grishman 2004) are designed to extract any relation expressed in text. Therefore, this paradigm enables large-scale extraction with no relation-specific training. Our relation extraction method is inspired by the seminal work of TextRunner (Banko and Etzioni 2008). TextRunner learns a CRF model, called O-CRF, to recognize tokens describing a relation between a pair of entities. O-CRF relies on relation-independent features, such as stop words and part-of-speech tags. In addition, TextRunner uses a self-supervision method to train O-CRF and a assessor module to prune out statements extracted with low confidence. Our method extends O-CRF in two ways: (1) by considering relations between entities and statements and (2) by using syntactic features found in parse and dependency trees.

Knowledge Representation. Knowledge representation models, such as Resource Description Framework (RDF) (W3C 2010), have long been able to represent meta statements. However, knowledge base extraction methods often use meta statements to store metadata about statements. For example, Yago stores the extraction date and a confidence score for a statement (Suchanek, Kasneci, and Weikum 2007). To the best of our knowledge, this is the first study to consider the problem of extracting meta statements from natural language documents.

3 Extracting Meta Statements

In this section we discuss our method to extract meta statements from blog posts.

Pre-processing Blog Posts. We process each post from the dataset separately, as follows. First, we identify sentence boundaries using LingPipe[1] and convert each such sentence into plain (ASCII) text for easier manipulation. (In the process, HTML tags and entities referring to special characters and punctuation marks are dealt with); this is accomplished with the Apache Commons library[2] and Unicode characters are converted into ASCII using the LVG component of the SPECIALIST library[3]).

Next, we identify entities in each sentence, using the LBJ Tagger[4], a state-of-the-art named entity recognition (NER) system (Ratinov and Roth 2009). LBJ assigns one of four categories (PER, ORG, LOC or MISC) to each entity it identifies. The final step is to identify names that refer to the same real-world entity. This is accomplished using a *coreference* resolution tool to group these names together. We used Orthomatcher from the GATE framework[5], which has been shown experimentally to yield very high precision (0.96) and recall (0.93) on news stories (Bontcheva et al. 2002). Observe that the coreference resolution is performed for entities within a blog post only.

Once we process all blog posts as described above, each sentence is then split into tokens using the OpenNLP library[6]. We explain our approach using the following sentence tokens (separated by white spaces):

> The *U.S.* is seeking ways to punish *Moscow* in response to *Russia*'s conflict with *Georgia* .

input: Sequence of tokens T
output: Set of statements S

```
1  S ← ∅;
2  A ← all entities mentioned in T;
3  P ← A × A ; // We consider every pair of arguments
4  foreach ⟨a1, a2⟩ ∈ P do
5      if m1 precedes m2 in T then
6          relation ← meta-CRF (T, m1, m2);
7          if relation is defined then
8              S ← S ∪ {(m1, relation, m2)};
9              a' ← sequence of tokens in T containing
               relation, m1 and m2;
               /* Remember the newly found argument */
10             P ← P ∪ (a' × A) ∪ (A × a');
11             A ← A ∪ a';
12         end
13     end
14 end
15 return S
```

Figure 2: Algorithm for finding statements and meta statements.

Notation. We represent named entities with italics, and relational terms with small capitalized letters. A statement is denoted by a triple of the form (arg_1, REL, arg_2), where REL is a relation, and arg_1 and arg_2 are the *arguments* of this relation. An argument can be either a named entity or another statement triple. A statement containing entities in its arguments is called *direct* statement. Conversely, a statement containing another statement as one of its arguments is called a *meta statement*.

In our example, the statements are:

s_1: (*US*, TO PUNISH, *Moscow*),
s_2: (*Russia*, CONFLICT WITH, *Georgia*),
s_3: (s_1, RESPONSE TO, s_2).

3.1 The Algorithm

Our algorithm (Figure 2) operates at the argument level, seamlessly considering both atomic arguments (i.e., entities) and triples (i.e., other statements). This is achieved as follows. On a first pass over the sequence of tokens given as input, we first identify all explicit mentions to entities and add them the set A which keeps all arguments in the text

[1] http://alias-i.com/lingpipe
[2] http://commons.apache.org/lang/
[3] http://lexsrv3.nlm.nih.gov/SPECIALIST/
[4] http://l2r.cs.uiuc.edu/~cogcomp/software.php
[5] http://gate.ac.uk/
[6] http://opennlp.sourceforge.net

(line 2). In our running example, this first step would result in

$A = \{$*U.S.*, *Moscow*, *Russia*, *Georgia*$\}$

The algorithm attempts to find all possible relations involving arguments that appear together in a single sentence (loop 4–14). Thus, for every pair (a_1, a_2) of arguments, such that a_1 precedes a_2 in a sentence we attempt to detect a valid relation (line 6). If such a relation $s = (a_1, \text{REL}, a_2)$ is found (as detailed in the next section), then we: (1) add s to the set of statements (line 8), and (2) create a new argument for s, for future consideration with other arguments already identified (lines 10, 11).

In our example, the first statements to be extracted are:

s_1: (*U.S.*, TO PUNISH, *Moscow*)

s_2: (*Russia*, CONFLICT WITH, *Georgia*)

Once they are added to both A and P, we have then:

$A = \{$*U.S.*, *Moscow*, *Russia*, *Georgia*, s_1, $s_2\}$

Thus, the algorithm will eventually attempt to identify (meta) statements involving the atomic entities and s_1 and s_2, thus producing s_3 above. It can be shown that the algorithm will never consider the same pair of arguments (atomic or otherwise) more than once, and thus always terminates.

3.2 The meta-CRF Model

In this section we discuss how to extract a relation between a pair of arguments from a sentence.

We model relation extraction as a sequence labeling problem — given a input sequence of tokens $\mathbf{x} = x_1, \ldots, x_n$, produce an output sequence of labels $\mathbf{y} = y_1, \ldots, y_n$ from a set of labels. In particular, we consider tokens *in between* two arguments and labels indicating whether a token belongs to a relation or not. We adopt the BIO encoding, a widely-used technique in natural language processing (Jurafsky and Martin 2008). This encoding marks the Beginning, Inside and Outside of a phrase; therefore, each token is labeled as B-REL, I-REL or O-REL. Figure 3 illustrates the tokens appearing in between "U.S." and "Moscow" and their respective labels. Tokens that should be labelled as B-REL or I-REL are called *relational* tokens.

Our method, called meta-CRF, is based on Conditional Random Fields (CRF) (Lafferty, McCallum, and Pereira 2001). CRF is a graphical model that estimates a conditional probability distribution, denoted $p(\mathbf{y}|\mathbf{x})$, over label sequence $\mathbf{y}$ given the token sequence $\mathbf{x}$. The probability of a label be assigned to the i-th token is defined by a vector $\mathbf{f} = \{f^1, f^2, \ldots, f^K\}$ of real-valued *feature functions* of the form $f^k(y_i, y_{i-1}, \mathbf{x}, i)$. Therefore, a feature function can be defined over the current label y_i, the previous label y_{i-1} or any token in $\mathbf{x}$. Examples of feature functions are:

$$f^1(y_i, y_{i-1}, \mathbf{x}, i) = [[x_i \text{ is an adverb}]].[[y_i = \text{O-REL}]]$$
$$f^2(y_i, y_{i-1}, \mathbf{x}, i) = [[x_i \text{ is a verb}]].[[y_i = \text{B-REL}]].[[y_{i-1} = \text{O-REL}]]$$

where the indicator function $[[condition]] = 1$ if $condition$ is true and zero otherwise. Each feature function f^k is associated with a weight W_k; therefore, there is a weight vector

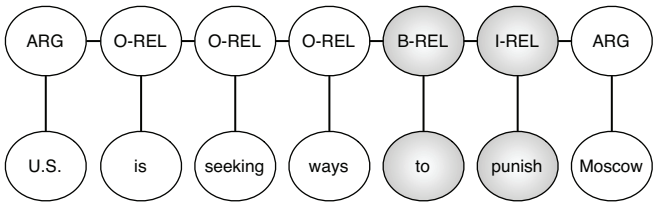


Figure 3: A CRF model is used to recognize the relation TO PUNISH between "U.S." and "Moscow".

$\mathbf{W} = W_1, \ldots, W_K$ corresponding to $\mathbf{f}$. Finally, we can define the CRF model as follows:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} e^{\mathbf{W}.\mathbf{F}(\mathbf{x},\mathbf{y})}$$

where $\mathbf{F}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{|\mathbf{x}|} \mathbf{f}(y_i, y_{i-1}, \mathbf{x}, i)$ and $Z(\mathbf{x})$ is a normalizing constant equal to $\sum'_{\mathbf{y}} e^{\mathbf{W}.\mathbf{F}(\mathbf{x},\mathbf{y}')}$.

Training meta-CRF consists in learning the weight vector $\mathbf{W}$. This vector defines the likelihood of associating a label to a individual token as well as transitioning from label to label. Meta-CRF uses the CRF implementation provided by the MALLET library (Mccallum 2002).

3.3 Features

The set of features used by meta-CRF is similar to those used by state-of-the-art relation extraction systems (Jurafsky and Martin 2008). We use tokens appearing between arguments, their part of speech, the argument types (statement or entity) and syntactic features from the parse and dependency tree.

Tokens. Following the OIE paradigm, we include as features the actual tokens belonging to *closed classes* (e.g. prepositions and determiners) but not function words such as verbs or nouns. For example, the tokens used from the sentence "AP reported Russia's conflict with Georgia" are "'s" and "with" only. This is because our method is designed to extract any relation not a specific one.

Part of speech. Every token is associated with its part of speech. Intuitively, we expect that relations in English follow a limited number of part-of-speech patterns. Banko and Etzioni present a study shows that 95% the relations in their dataset follow eight simple part-of-speech patterns. An example is "settlement with", which follows the pattern: noun→preposition. Figure 4 presents the tokens (in bold) from the sentence "AP reported Russia's conflict with Georgia" along their part-of-speech tags (in italics).

Argument Type. Instead of simply assigning the label "ARG" to arguments, we assign a label that corresponds to the type of the argument ("ENTITY" or "STATEMENT").

Parse tree. Our method uses the path length between a token and each argument in a full parse tree. Intuitively, we expect that the paths between relational tokens and their arguments to be relatively short. The node representing an

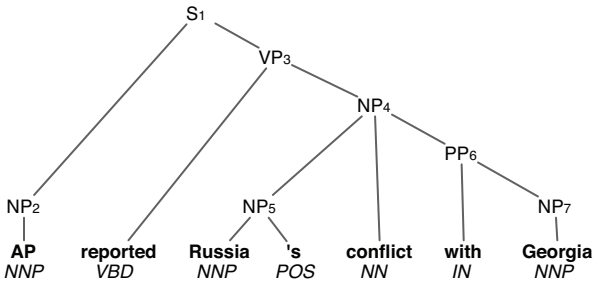


Figure 4: Parse tree for the sentence "AP reported Russia's conflict with Georgia" following the Penn TreeBank terminology (Bies 1995). Non terminals are numbered for easy identification. Tokens are highlighted in bold and part-of-speech tags are in italic.

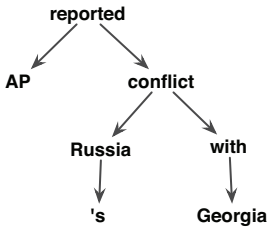


Figure 5: Dependency tree for the sentence "AP reported Russia's conflict with Georgia".

argument is the lowest common ancestor of all tokens in that argument. Figure 4 gives an example parse tree. The (atomic) arguments "AP", "Russia" and "Russia's conflict with Georgia" are represented by the nodes NP_2, NP_5 and NP_4, respectively. Observe that the path between NP_2 and NP_5 (NP_2–S_1–VP_3–NP_4–NP_5) is longer than the path between NP_2 and NP_4 (NP_2–S_1–VP_3–NP_4), indicating that "AP" and "Russia's conflict with Georgia" are more likely to form a statement than "AP" and "Russia" alone. Our method generates a parse tree for each sentence by using the tools available from OpenNLP.

Dependency tree. We also use the path length between a token and each argument in a dependency tree. Intuitively, shorter paths are likely to indicate stronger dependency between the tokens and the arguments. Figure 5 illustrates an example dependency tree. An argument is represented by the root of the minimal subtree containing all its tokens. For example, "Russia's conflict with Georgia" is represented by "conflict". Observe that the path between "AP" and "Russia" (AP–reported–conflict–Russia) is longer than the path between "AP" and "Russia's conflict with Georgia" (AP–reported–conflict). Our method produces a dependency tree for each sentence by applying the algorithm from Xia and Palmer (2001).

The limitations of meta-CRF are shared with most relation extraction systems. Our method focuses in relations that are explicitly expressed in the text, and not implied by punctuation or structural clues, for example. In addition, relations must appear in the text between arguments. Banko and Etzioni (2008) present a study showing that more than 80% of binary relations are found in the text window between arguments, as oppose to the windows before and after the pair of arguments. Finally, our method focuses on relations expressed within a sentence as oppose to relations that cross sentence boundaries, such as in "Russia invaded Georgia. U.S. condemned the invasion."

3.4 The need for syntactic features

State-of-the-art relation extraction systems based on CRF, such as TextRunner, often rely almost exclusively on part-of-speech tags. One problem with this approach is that part-of-speech tags and other morphologic features are insufficient for dealing with meta statements in the text. To see this, consider the sentence "AP reported Russia's conflict with Georgia" and its parse tree illustrated in Figure 4. Observe that "AP" and "Russia's conflict with Georgia" presents "reported" as a relation between them. Furthermore, observe that "AP" and "Russia" also contains "reported" between them, but in this case "reported" does not represent a relation.

In both cases, the part of speech sequence is the same: ARGUMENT $\rightarrow$ VBD $\rightarrow$ ARGUMENT. Therefore, a CRF model has no choice but to assign the same label to "reported" in both cases. No matter the label assigned by the model, this label will be incorrect for at least one of the above argument pairs. This lose-lose situation is very common when dealing with meta statements, since statement arguments will always contain entity arguments.

Our solution for the above problem is to rely on the syntactic structure of a sentence. Parse and dependency trees often provide useful hints to determine whether a sentence presents a relation between two arguments or not. As discussed in Section 3.3, we observe that the path between "AP" and "Russia" is longer than the path between "AP" and "Russia's conflict with Georgia" in both parse and dependency trees. Our observations are in agreement with a recent study that claims that relations can be extracted almost exclusively from the path between arguments in a dependency tree (Bunescu and Mooney 2005).

4 Experimental Validation

In this section we present the results of an experimental evaluation of meta-CRF. Our method uses all features described in Section 3.3. We use as baseline a CRF model that relies on the features used by TextRunner's O-CRF (tokens and their part of speech).

4.1 Setup

Our experiments use sentences from the ICWSM Spinn3r blog dataset (Burton, Java, and Soboroff 2009). The ICWSM dataset contains 25 million English posts published between August 1st and October 1st, 2008. Popular topics include the 2008 U.S. Election, the Russian conflict with Georgia, the Olympics and the economic crisis. We manually collected a hundred sentences from blog posts containing popular entities in politics (e.g., Barack Obama, John

Unit	Quantity
Original Sentences	100
Examples	496
Meta statements	107
Direct statements	111
No statement	278
Tokens	1245
Relational tokens	364
Non relational tokens	881

Figure 6: Details about the examples used in experiments. "Original sentences" indicates the sentence collected from the ICWSM dataset, "Examples" are sentences annotated with arguments and relations (containing meta statements, direct statements and no statements). "Tokens" indicates the number of tokens in all examples. "Relational tokens" indicate tokens labeled as relations (B-REL, I-REL) and "Non relational tokens" indicate tokens labeled as O-REL.

McCain), sports (e.g., Michael Phelps), entertainment (e.g., Paris Hilton) and entities involved in the conflict in Georgia (e.g., Russia, U.S.).

Each collected sentence was used to produce positive (containing direct and meta statements) and negative examples (containing no statements). We produce an example for each pair of arguments in a sentence. For example, the examples produced from "U.S. condemned Russia's conflict with Georgia" are:

U.S. CONDEMNED *Russia's conflict with Georgia*

U.S. condemned *Russia* 's CONFLICT WITH *Georgia*

U.S. condemned *Russia* 's conflict with Georgia

U.S. condemned Russia's conflict with *Georgia*

where tokens in italics are arguments and small capitalized tokens indicate relations. Observe that the first example contains a meta statement, the second example contains a direct statement and the last two contain no statements. Producing examples in this way may result in many negative examples where arguments are separated by many tokens. To limit the number of negative examples, we prune out every argument pair where the arguments are separated by more than 5 tokens.

We use both positive and negative examples to evaluate our method. Figure 6 provides more information about these examples. Our experiments rely on tenfold cross validation by splitting the examples into ten partitions. In each round, a partition is used for testing while the nine remaining are used for training.

Metrics. The quality of the extraction is measured by the number of tokens correctly labeled. The extraction accuracy is defined as follows.

$$\text{Accuracy} = \frac{\text{Correct Labels}}{\text{Number of Tokens}}$$

Round	O-CRF	meta-CRF	Improvement
1	0.78	0.89	14.4%
2	0.75	0.89	17.7%
3	0.77	0.89	14.6%
4	0.72	0.91	25.6%
5	0.69	0.85	22.7%
6	0.71	0.83	16.3%
7	0.70	0.79	11.6%
8	0.67	0.89	34.1%
9	0.63	0.77	21.5%
10	0.68	0.85	25.0%
Average	0.71	0.86	20.1%

Figure 7: Results for O-CRF and meta-CRF in each round of a tenfold cross-validation evaluation. "Improvement" indicates the relative gain in performance by meta-CRF over O-CRF.

4.2 Comparison between O-CRF and meta-CRF

We use O-CRF as our baseline for comparison as it is the state-of-the-art of CRF-based relation extraction methods. It should be noted that while O-CRF is not a method for extracting meta statements (nor their authors claim so), this comparison is valuable in that it provides an objective way to assess the impact of using syntactic features when extracting meta statements, as oppose to relying almost completely on part-of-speech tags.

Figure 7 presents the accuracy results for O-CRF and meta-CRF in each experimental round. Observe that meta-CRF improves O-CRF performance by over 20% on average. In addition, our method consistently outperforms O-CRF in every round with a minimum improvement of 11.6% and maximum improvement of 34.1%.

Figure 8 details the performance of meta-CRF and O-CRF by reporting their results on examples that contain meta statements, direct statements and no statements in separate. Observe that our method almost tripled the results obtained by O-CRF when extracting meta statements. Figure 8 also shows that our method almost doubled O-CRF performance on examples containing direct statements. This result can be explained by our method's ability to better differentiate direct and meta statements by using structural information as explained in Section 3.4. The lack of syntactic information led O-CRF to label most relational tokens as non relational. An in-depth investigation revealed that O-CRF was able to correctly label relation tokens only 21% of the time (a metric known as recall), while our method reported 78% at the same task. This is because many examples present the same part-of-speech tag sequence but different labels (recall Section 3.3). O-CRF's inclination to label tokens as O-REL also explains why our method was unable to improve O-CRF performance at labelling non relational tokens when compared to our method (3.2% drop). Since O-REL comprises the majority of labels in our example set, the meta-CRF overall improvement (20.1%) was substantially below the improvement in examples containing direct (189.7%) and meta examples (82.4%).

	O-CRF	meta-CRF	Improvement
Meta statement	0.271	0.785	189.7%
Direct statement	0.4392	0.801	82.4%
No statement	0.9259	0.8965	-3.2%
All examples	0.71	0.86	20.1%

Figure 8: The performance of O-CRF and meta-CRF on average for examples containing meta statements, direct statements and no statements. "Improvement" indicates the relative gain in performance by meta-CRF over O-CRF.

Method	Accuracy	Improvement
O-CRF	0.71	–
+ Argument types	0.82	14.8%
+ Dependency	0.81	14.2%
+ Parse Tree	0.80	12.2%
All Features	0.86	20.1%

Figure 9: Impact of extending O-CRF with individual features. "+ Feature" indicates the model O-CRF extended with "Feature".

4.3 Contribution of Individual Features

In this experiment our goal is to study the contribution of individual features to our method's overall performance. Figure 9 shows the results for our baseline extended with the following features: argument types, dependency tree and parse tree. Observe that all features combined outperformed individual features. Furthermore, the addition of each individual features produces better accuracy than our baseline.

Another interesting result is that relying on dependency trees yields results as good as those obtained considering argument types alone, which explicitly provide weather an argument is an entity or a statement. This result shows the discriminative power of a dependency tree to differentiate between direct and meta statements.

5 Conclusion

This paper discussed a method for extracting reified information networks from natural language text, and results of applying this method to the ICWSM Spinn3r blog dataset. Unlike previous work, we focus on the simultaneous extraction of both direct statements, connecting entities, as well as meta statements, connecting entities and/or other statements. We proposed meta-CRF, a CRF-based model that extracts both direct and meta statements seamlessly. Our model extends TextRunner's O-CRF model by also incorporating syntactic features as found in parse and dependency trees. We showed the need for these syntactic features when dealing with meta statements. Finally, our evaluation reported that meta-CRF outperforms O-CRF by as much as 20% at extracting relations from a sample of the blogosphere.

5.1 Discussion

Overall, our meta-CRF method was able to extract meta statements with 0.86 accuracy, which, alone, is already satisfactory for information extraction tasks (Sarawagi 2008). Also, meta-CRF improved the state-of-the-art by over 20%. These results indicate, in a sense, the limitation of relying mainly on part-of-speech tags to extract meta statements. The root of this limitation is that, with O-CRF, one cannot avoid positive and negative examples which have the exact same features (recall our example on Section 3.3).

It is worth mentioning that this confusion introduced into O-CRF is unavoidable, and not an artifact of the way in which we train the models. In fact, our examples were produced automatically from pairs of arguments in each sentence. Also, we tried to achieve the standard 50/50 split between positive (218) and negative examples (278) by automatically pruning some of the negative examples (recall Section 4.1). Since negative examples are necessary to properly train a CRF model, it is thus hard to see a way of avoiding this confusion with O-CRF.

5.2 Future Work

To conclude, we provide some ideas for future work.

Our results indicate that meta-CRF often outperformed O-CRF even when extracting direct statements only. This happened, for instance, on sentences such as "AP reported Russia's conflict with Georgia", where we observed that a method needs to, at least, detect the meta statement involving AP and the conflict. By doing so, the method avoids extracting spurious relations, such as:

(*AP*, REPORTED, *Russia*).

This improvement over O-CRF indicates that our model might be useful in an industry-strength information extraction system such as TextRunner. It would be interesting, for instance, to investigate whether the self-supervised training method used in TextRunner can be applied to our model.

Our method's improvement over O-CRF comes at expense of processing time. This is because parse and dependency trees require heavyweight full parsing techniques. Processing time is a real concern when dealing with large amounts of text as found in the blogosphere. In these cases, shallow parsing is often adopted as a lightweight alternative. Therefore, we plan to investigate the effectiveness-efficiency tradeoff of using shallow parsing rather than full parsing.

Acknowledgements. This work was supported in part by a grant from the NSERC Business Intelligence Network.

References

Agichtein, E., and Gravano, L. 2000. Snowball: extracting relations from large plain-text collections. In *Proceedings of the ACM Conference on Digital libraries*, 85–94. ACM.

Banko, M., and Etzioni, O. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 28–36. Columbus, Ohio: Association for Computational Linguistics.

Bies, A. 1995. Bracketing Guidelines for Treebank II Style Penn Treebank Project.

Bollegala, D. T.; Matsuo, Y.; and Ishizuka, M. 2010. Relational duality: unsupervised extraction of semantic relations

between entities on the web. In *Proceedings of the International Conference on World Wide Web*, 151–160. New York, NY, USA: ACM.

Bontcheva, K.; Dimitrov, M.; Maynard, D.; Tablan, V.; and Cunningham, H. 2002. Shallow methods for named entity coreference resolution. In *Chaines de references et resolveurs d'anaphores, workshop TALN*.

Brin, S. 1998. Extracting patterns and relations from the world wide web. In *Proceedings of the World Wide Web and Databases (WebDB) International Workshop*, 172–183.

Bunescu, R. C., and Mooney, R. J. 2005. A shortest path dependency kernel for relation extraction. In Mooney, R. J., ed., *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 724–731. Association for Computational Linguistics.

Burton, K.; Java, A.; and Soboroff, I. 2009. The icwsm 2009 spinn3r dataset. In *Proceedings of the Annual Conference on Weblogs and Social Media (ICWSM)*.

Culotta, A., and Sorensen, J. S. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 423–429. Association for Computational Linguistics.

Etzioni, O.; Cafarella, M.; Downey, D.; Kok, S.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 2004. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the International conference on World Wide Web*, 100–110. New York, NY, USA: ACM.

GuoDong, Z.; Jian, S.; Jie, Z.; and Min, Z. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 427–434. Association for Computational Linguistics.

Hasegawa, T.; Sekine, S.; and Grishman, R. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 415. Association for Computational Linguistics.

Jurafsky, D., and Martin, J. H. 2008. *Speech and Language Processing*. Prentice Hall, 2 edition.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*, 282–289. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Mccallum, A. K. 2002. MALLET: A Machine Learning for Language Toolkit.

McDonald, R.; Pereira, F.; Kulick, S.; Winters, S.; Jin, Y.; and White, P. 2005. Simple algorithms for complex relation extraction with applications to biomedical IE. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 491–498. Association for Computational Linguistics.

Mesquita, F.; Merhav, Y.; and Barbosa, D. 2010. Extracting information networks from the blogosphere: State-of-the-art and challenges. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM), Data Challenge Workshop*.

Ratinov, L., and Roth, D. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Conference on Computational Natural Language Learning*, 147–155. Association for Computational Linguistics.

Sarawagi, S. 2008. Information extraction. *Found. Trends databases* 1:261–377.

Suchanek, F.; Kasneci, G.; and Weikum, G. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, 697–706. ACM.

W3C. 2010. RDF Primer.
`http://www.w3.org/TR/rdf-primer/`.

Wick, M.; Culotta, A.; and McCallum, A. 2006. Learning field compatibilities to extract database records from unstructured text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 603–611. Association for Computational Linguistics.

Xia, F., and Palmer, M. 2001. Converting dependency structures to phrase structures. In *Proceedings of the International Conference on Human Language Technology Research*, HLT '01, 1–5. Stroudsburg, PA, USA: Association for Computational Linguistics.

Xu, F.; Uszkoreit, H.; and Li, H. 2007. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Yang, G., and Kifer, M. 2003. Reasoning about anonymous resources and meta statements on the semantic web. *Journal on Data Semantics* 69–97.

Zelenko, D.; Aone, C.; and Richardella, A. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.* 3:1083–1106.

Zhu, J.; Nie, Z.; Liu, X.; Zhang, B.; and Wen, J.-R. 2009. Statsnowball: a statistical approach to extracting entity relationships. In *Proceedings of the International Conference on World Wide Web*, 101–110. ACM.