Extracting Information Networks from the Blogosphere

YUVAL MERHAV, Illinois Institute of Technology
FILIPE MESQUITA and DENILSON BARBOSA, University of Alberta
WAI GEN YEE, Orbitz Worldwide
OPHIR FRIEDER, Georgetown University

We study the problem of automatically extracting information networks formed by recognizable entities as well as relations among them from social media sites. Our approach consists of using state-of-the-art natural language processing tools to identify entities and extract sentences that relate such entities, followed by using text-clustering algorithms to identify the relations within the information network. We propose a new term-weighting scheme that significantly improves on the state-of-the-art in the task of relation extraction, both when used in conjunction with the standard $tf \cdot idf$ scheme and also when used as a pruning filter. We describe an effective method for identifying benchmarks for open information extraction that relies on a curated online database that is comparable to the hand-crafted evaluation datasets in the literature. From this benchmark, we derive a much larger dataset which mimics realistic conditions for the task of open information extraction. We report on extensive experiments on both datasets, which not only shed light on the accuracy levels achieved by state-of-the-art open information extraction tools, but also on how to tune such tools for better results.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: open information extraction, relation extraction, named entities, domain frequency, clustering

ACM Reference Format:
Merhav, Y., Mesquita, F., Barbosa, D., Yee., W. G., and Frieder, O. 2012. Extracting information networks from the blogosphere. ACM Trans. Web 6, 3, Article 11 (September 2012), 33 pages.
DOI = 10.1145/2344416.2344418 http://doi.acm.org/10.1145/2344416.2344418

1. INTRODUCTION

The extraction of structured information from text is a long-standing challenge in natural language processing, which has been reinvigorated by the ever-increasing

A preliminary version of this paper appeared in the AAAI ICWSM 2010 Data Challenge Workshop.
The authors acknowledge the generous support of the Natural Sciences and Engineering Research Council of Canada through BIN–the Business Intelligence Network, and Alberta Innovates.
Part of this work was done while Y. Merhav was a research intern at the Computing Science Department, University of Alberta, Canada.
Authors' addresses: Y. Merhav, Information Retrieval Laboratory, Computer Science Department, 10 West 31st Street, Stuart Building 235, Chicago, IL 60616; email: ymerhav@iit.edu; F. Mesquita and D. Barbosa, Computing Science Department, 2-32 Athabasca Hall, University of Alberta; email: mesquita, denilson@cs.ualberta.ca; Wai Gen Yee, Orbitz Worldwide, 500 W. Madison St., Suite 1000, Chicago, IL 60661; email: wyee@orbitz.com; O. Frieder, Computer Science Department, 347 St. Mary's Hall, Washington, D.C. 20057-1232; email: ophir@cs.georgetown.edu.

DOI 10.1145/2344416.2344418 http://doi.acm.org/10.1145/2344416.2344418

availability of user-generated textual content online. One environment that stands out as a source of invaluable information is the *blogosphere*–the network of social media sites, in which individuals express and discuss opinions, facts, events, and ideas pertaining to their own lives, their communities, professions, or societies at large. Indeed, the automatic extraction of reliable information from the blogosphere promises a viable approach for discovering very rich social data—the issues that engage society in thousands of collective and parallel conversations online. Furthermore, as the blogosphere attracts more and more participants from all segments of society, the extraction of information networks from the blogosphere will provide a better understanding of our collective view of the society we live in and talk about.

Considerable attention has been paid to the problem of automatically extracting and studying social dynamics among the participants (i.e., authors) in shared environments like the blogosphere. Unlike them, our goal is to extract entities, facts, ideas, and opinions, as well as the relationships among them, which are expressed and discussed collectively by blog authors. Such structured data can be organized as one or more *information networks*, which in turn are powerful metaphors for the study and visualization of various kinds of complex systems [Knox et al. 2006]. Figure 1 shows an example of such a network, in this case the *egocentric* network [Hanneman and Riddle 2005] around the entity "Barack Obama".[1] This network was built with data extracted from blog posts collected between August and September of 2008, before the United States presidential elections. The self-evident power of the network in Figure 1 to illustrate the discussions in the blogosphere is very compelling: it accurately shows the important entities discussed during the election and the most prominent relations amongst them. The figure also shows some unexpected connections, such as Britney Spears and Paris Hilton; they were used in a campaign advertisement by John McCain, who tried to associate Barack Obama with the two celebrities who "are often portrayed as frivolous and irresponsible" [CNN 2008].

1.1. Problem Definition

We assume a set E of unique *entities* in the network, each represented as a $\langle name, type \rangle$-pair. We assume a set T of *entity types*, which are usually automatically assigned to each recognized entity; in our work we consider the types `PER` (for Person), `ORG` (Organization), `LOC` (Location) or `MISC` (Miscellaneous).

An edge (e_1, e_2, l) in the network represents the *relationship* between entities e_1, e_2 identified by the label l, such as

$$r = (\langle \text{Barack Obama, PER} \rangle, \langle \text{John McCain, PER} \rangle, \text{opponent}).$$

The *domain* of a relationship is defined by the types of the entities in it; for instance, the domain of r above is `PER`–`PER`. A *relation* consists of the set of all edges that have the same label. We call a relation *homogeneous* if all its pairs have the same domain. Finally, a *network* consists of a set of entities and a set of relations involving such entities.

Identifying the relationship (if any) between entities e_1, e_2 is done by analyzing the sentences that mention e_1 and e_2, together. An *entity pair* is defined by two entities e_1 and e_2, together with the *context* in which they co-occur. For our purposes, the context can be any textual feature that allows the identification of the relationship for the given pair. As an illustration, Table I shows entity pairs where the context consists of the exact text *in between* the mentioned entities. As we will discuss later, we actually employ standard information retrieval techniques to extract the context from the text

[1]This network was automatically extracted by our tool from a large sample of the blogosphere; the analysis and visualization of the network was done with NodeXL (http://nodexl.codeplex.com/).

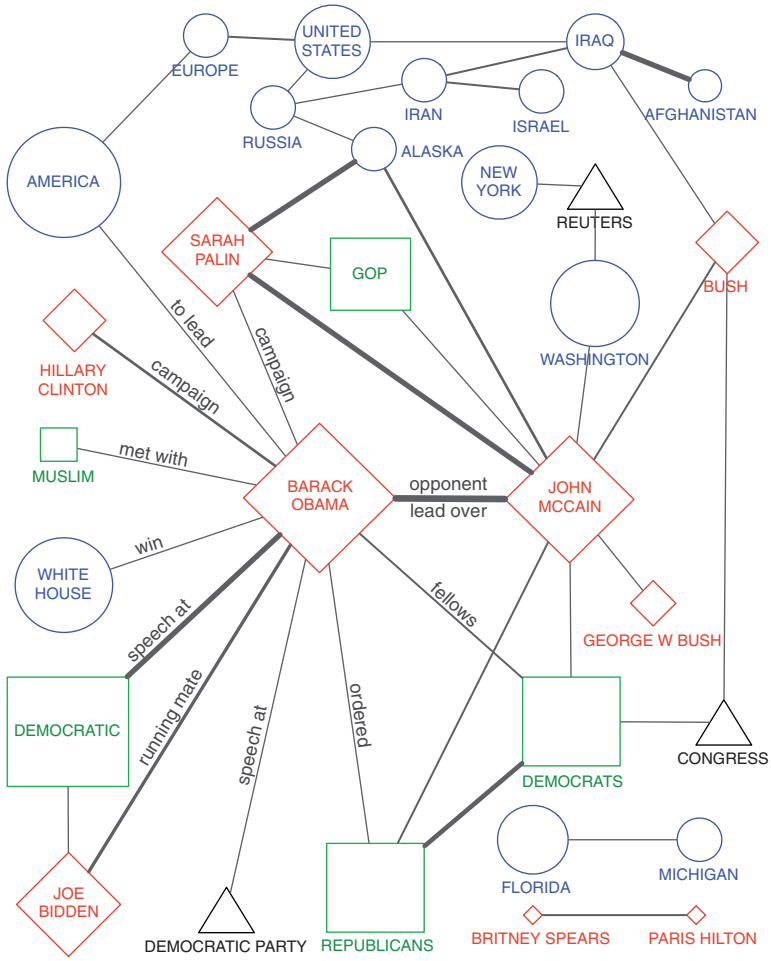


Fig. 1. Egocentric perspective of the information network extracted by SONEX from the Spinn3r dataset, focusing on Barack Obama. Shapes and colors represent entity types: red diamonds indicate persons, black triangles indicate organizations, blue circles indicate locations, and green squares indicate miscellaneous. The size of the nodes indicate their centrality in the network, and the width of the edges indicate their support, measured by the number of sentences that express that relation. For clarity, only edges with the highest support are shown.

Table I. Entity Pairs from the Spinn3r Dataset

Entity 1	Context	Entity 2
⟨Barack Obama, PER⟩	and vice presidential running mate and his running mate Sen. received only a minor fundraising bump after he named	⟨Joe Biden, PER⟩
⟨John McCain, PER⟩	running mate has chosen as his running mate apparently even didn't bother Googling	⟨Sarah Palin, PER⟩

in the sentences. As with a relationship, the *domain* of a pair consists of the types of the entities in that pair. The *popularity* (or support) of an entity pair is defined by the number of sentences connecting the two entities.

Problem definition. We can now define our work more precisely. Given a collection of documents (blog posts in the work described in this article), we extract an information network in a fully unsupervised way. This problem has been termed an open information extraction (OIE) in the literature [Banko et al. 2007].

1.2. Challenges

Many challenges exist in developing an OIE solution. First, recognizing and disambiguating entities in a multidocument setting remains a difficult task [Jurafsky and Martin 2009]. Second, the unsupervised nature of the problem means that we have to identify all relations from the text only. This is done by identifying so-called *relational* terms in the sentences connecting pairs of entities. Relational terms are words (usually one or two) that describe a relation between entities (for instance, terms like "running mate," "opponent," "governor of" are relational terms, while "fundraising" and "Googling" are not). Finally, another massive challenge is that of evaluating the resulting relations extracted by the OIE system: as discussed further below, the state-of-the-art in the literature relies on small-scale benchmarks and/or manual evaluation. However, neither approach applies to the domain we address (blogs).

It is worth mentioning that, besides the technical challenges mentioned above, there are other practical issues that must be overcome if we want to deploy any OIE system in the blogosphere. For instance, often, bloggers copy text from each other, leaving a high number of duplicate content. This, in turn, introduces considerable bias in the final network extracted by the system. One common solution is to work on distinct sentences. Also, most algorithms involved in OIE are computationally intensive, and considerable engineering is required to arrive at practical systems.

1.3. Outline and Contributions

Our OIE solution, SONEX (SOcial Network EXtractor), works as follows. First, we use a state-of-the-art named entity recognition (NER) to extract all entities mentioned in the document collection. Next, we extract all sentences from the text that mention two entities, and from those, extract all entity pairs. Then, we use a clustering algorithm to group similar pairs together. Finally, we find a representative term (usually one or two words) from each cluster and assign it as the relation label. SONEX is completely self-contained, not relying on any external knowledge base, which makes it suited for dynamic environments.

We deployed SONEX on the ICWSM 2009 Spinn3r corpus [Burton et al. 2009], focusing on posts in English (25 million out of 44 million in total), collected between August 1st, 2008 and October 1st, 2008. The total size of the corpus is 142 GB (uncompressed). It spans a number of big news events (e.g., 2008 Olympics, US presidential election, the beginning of the financial crisis), as well as everything else we might expect to find in blog posts. SONEX runs in a distributed fashion, lending itself as a highly scalable solution: using 10 commodity desktop PCs, we were able to extract entity pairs from 10 million blog posts per day.

SONEX builds on state-of-the-art text-clustering methods to group the entity pairs into (un-labeled) relations. We tested various algorithms, using different textual features for the context for the entity pairs. We observed that the customary $tf \cdot idf$ weighting scheme (which is used in the state-of-the-art) is often suboptimal in the task of relation extraction, as it does not take into account the context in which a relation is defined. Thus, we use a novel weighting scheme that combines $tf \cdot idf$ with what we

call the *domain frequency* (df), which exploits semantic information about the relations being extracted. We show that our new weighting scheme outperforms the state-of-the-art.

As for the evaluation of our system, we developed a method that exploits a curated database (Freebase in this work) to generate a benchmark that is specific to the Spinn3r corpus, suitable to evaluate the output of SONEX. Our resulting benchmark is comparable in size to the best hand-crafted ones described in the literature, but is (of course) restricted to the entity pairs that appear in the intersection of the curated database and the Spinn3r corpus. We complement this fully unsupervised evaluation with a manual evaluation that considers several thousands possible pairs, to assess the performance of SONEX on a more realistic scenario.

In summary, our contributions are as follows.

—We present the first large-scale study on using a text-clustering-based approach to OIE on the blogosphere, indicating promising results.
—We introduce a novel weighting scheme that outperforms the classical $tf \cdot idf$ in the task of relation extraction.
—We develop a fully automated and rigorous method for testing the accuracy of a relation-extraction system, tailored to a specific corpus.

2. RELATED WORK

2.1. Social Networks of the Blogosphere

The blogosphere has attracted many researchers who study social networks, as many of the social relationships between blog authors (bloggers) are explicitly stated in the form of links. For example, Marlow used links between blogs to construct the social networks of the blogosphere, and then employed social network analysis to describe the aggregate effects of status by means of popularity and influence [Marlow 2004]. In this work, we do not attempt to extract or analyze the social structure of blog authors; instead, we extract the information networks from the actual blog data. In particular, we aim at extracting relations between named entities cited in blog posts. Methods for this problem, known as relation extraction, follow one of the following paradigms: targeted and open information extraction. We discuss them in the following sections.

2.2. Targeted Information Extraction

Methods in this paradigm learn to extract a single predefined and domain-specific relation. Two approaches are prominent: bootstrapping and supervised learning. Bootstrapping methods use sample instances of a relation as input to successively extract more instances [Brin 1998; Agichtein and Gravano 2000]. On the other hand, methods using supervised learning exploit linguistic and statistical features defined a priori [Kambhatla 2004; GuoDong et al. 2005; Fisher et al. 1995; Craven et al. 2000; Rosario and Hearst 2004], and kernels methods [Zelenko et al. 2003; Bunescu and Mooney 2005; Culotta and Sorensen 2004]. Since the effort to provide training is linear to the number of relations, these methods are not designed to tackle massive corpora containing a large number of unknown relations [Banko et al. 2007]. In the next section, we discuss a paradigm that extracts relations without requiring any relation-specific training.

2.3. Open Information Extraction

The large-scale extraction of *unanticipated* relations has been termed as open information extraction (OIE) [Banko et al. 2007]. Since the relations are not known in advance, OIE also requires automatically assigning labels to each discovered relation. Recent systems addressing this problem can be divided into three main approaches:

bootstrapping, self-supervised, and unsupervised. Recent bootstrapping approaches include the StatSnowBall system that employs a general framework that iteratively extracts relational patterns based on initial seed, and weights patterns every iteration using l-normalization to extract only good patterns [Zhu et al. 2009]. Their main experiments on a private Web dataset were focused on two relation types, Husband and Wife. Bunescu and Mooney used a search engine to construct a large training set for a classifier, starting only with a small set of training data. Two data sets created by the authors were used for evaluation with two relation types, corporate acquisition, and person birth place [Bunescu and Mooney 2007].

The inspiring TextRunner [Banko et al. 2007] work is the first self-supervised approach for OIE on a Web scale. For each pair of noun phrases that is not filtered based on several constraints, TextRunner applies a self-supervised learner to train a naive-Bayes classifier. The learner is called self-supervised because it produces its own positive and negative examples of how relations are expressed in English. The learner applies a full syntactic parser on a number of sentences and constructs a dependency tree of each sentence; constructing a dependency tree is expensive and not practical in large datasets, which is the reason why TextRunner uses it only on a subset. TextRunner was evaluated on 10 relation types to compare it to a targeted system such as KnowItAll [Etzioni et al. 2004]. The authors also applied small manual evaluations and estimation techniques to evaluate the actual performance of the system on the entire relation set it extracted from the Web. Later on, TextRunner's extractor was improved by using conditional random fields [Banko and Etzioni 2008]. Other self-supervised works include WOE [Wu and Weld 2010] that utilized heuristic matches between Wikipedia Infobox attribute values and corresponding sentences to construct training data; they used three corpora for their experiments, WSJ from Penn Treebank, Wikipedia, and the general Web; however, for each data set only 300 sentences were used for evaluation. Mintz et al. also extracted relations from Wikipedia; they applied a distant supervision approach using Freebase as a distant source for automatic supervision, avoiding the domain dependence and small-scale of existing datasets that are used in supervised approaches [Mintz et al. 2009]. A recent rule-based system is ReVerb [Fader et al. 2011]. Reverb identifies candidate relations in a sentence by using a regular expression over part-of-speech tags. A confidence score for each candidate relation is then computed by a logistic regression method. Candidates with scores below a threshold are discarded. ReVerb has been shown to outperform both TextRunner and WOE in a manual evaluation over 500 sentences [Fader et al. 2011]. We perform a comparative experiment between SONEX and ReVerb, which shows that SONEX achieves much higher recall than ReVerb on our dataset.

Fully unsupervised OIE systems (such as ours) are mainly based on clustering of entity pairs context to produce relations, as introduced by Hasegawa et al. [2004]. Hasegawa et al. used single words (unigrams) to build the context vectors and applied hierarchical agglomerative clustering (HAC) with complete linkage since it produced better results than single and average linkage. Their evaluation was based on a 1995 New York Times corpus; they analyzed the data set manually and identified relations for two different domains: 177 distinct PER-LOC pairs and 65 distinct ORG-ORG pairs. The two sets were manually classified into 38 and 10 distinct relations, respectively. No labeling evaluation was applied. Zhang et al. used parse trees of the context to allow pairs to appear in more than one cluster [Zhang et al. 2005]. They assigned labels to clusters based on the most frequent head word (as defined by a deep linguistic parser) in a cluster. To reduce noise in the feature space—a common problem with text mining—known feature selection and ranking methods for clustering were applied [Chen et al. 2005; Rosenfeld and Feldman 2007]. Both works used the K-means clustering algorithm with the stability-based criterion to automatically estimate the

Fig. 2. Workflow for extracting annotated sentences from the actual blog posts.

number of clusters. Rosenfeld and Feldman reported that the HAC with single linkage outperformed both K-means and the other variants of HAC [Rosenfeld and Feldman 2007]. In contrast, in this work we report that both complete and average linkage outperformed single linkage. We believe that the different outcome is a result of different datasets (size, relations, domains) and evaluation process. Finally, Shinyama and Sekine first clustered the document collection into similar topics and then identified relations between entities within each cluster [Shinyama and Sekine 2006]. SONEX extends these works by clustering entity pairs extracted from the blogosphere using a novel weighting scheme. As far as we know, this is the first work to address the problem of relation extraction in this environment.

2.4. Evaluation

The current practice for evaluating accuracy of relation extraction systems resorts to using gold standard relations from the automatic content extraction (ACE)[2] or by clustering a small number of entity pairs manually as in Hasegawa et al. [2004] and [Rosenfeld and Feldman 2007; Bunescu and Mooney 2007]. The ACE RDC 2003 and 2004 benchmarks [Doddington et al. 2004] are private corpora composed by news articles; hence they are not suitable for evaluating relations extracted from the blogosphere. Blog posts are usually written for a more restricted audience; documents as such contain different writing characteristics than formal news articles written for a large audience [Minkov and Wang 2005]. On the other hand, it is hard to conduct an unbiased evaluation by choosing and clustering pairs manually. Furthermore, it is difficult to compare results from different works. One of our contributions are automatic methods for evaluating both the clustering and labeling processes.

Our evaluation method uses an external data source as ground truth for evaluating OIE systems. External data sources have been used in evaluation methods for traditional systems [Agichtein and Gravano 2000; Mintz et al. 2009]. However, these methods do not apply to OIE systems, since there is no trivial equivalence between the relations extracted by an OIE system and relations from a data source. Earlier evaluation methods for OIE systems are all based on a human-produced ground truth [Banko et al. 2007; Hasegawa et al. 2004; Fader et al. 2011]. Our evaluation (20 relations, 395 pairs with 45710 instances) has a comparable number of relations and pairs as those used in Hasegawa et al. (48 relations, 242 pairs).

3. EXTRACTING ENTITY PAIRS

The first step in SONEX is processing the blog posts in the corpus to obtain *annotated* sentences in which named entities are mentioned. From these sentences, we construct the entity pairs which are then clustered during the relation identification (discussed in the next section). Figure 2 illustrates the workflow for extracting the annotated sentences from the blog posts. The process starts with the identification of sentence boundaries (using LingPipe,[3]) followed by a conversion of each 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

[2]http://projects.ldc.upenn.edu/ace/.
[3]http://alias-i.com/lingpipe.

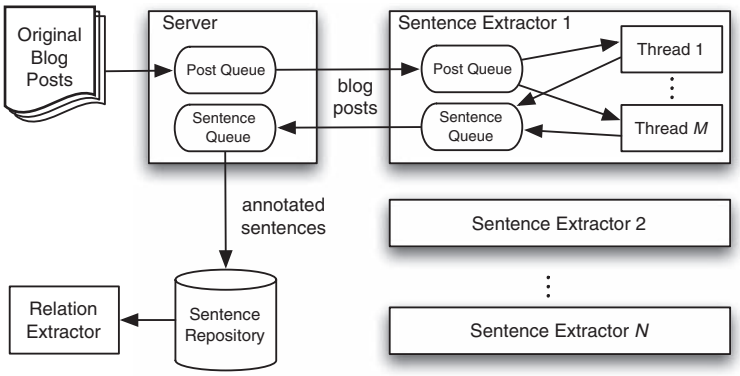


Fig. 3. SONEX architecture. The Server sends blog posts to entity extractors, which parse the posts and sends them back to the server to be saved in a database. The relation extractor takes parsed posts from the database and identifies relations between entities.

the Apache Commons library[4] and Unicode characters are converted into ASCII using the LVG component of the SPECIALIST library.[5])

The second step consists of identifying entities in each sentence and assigning their types. For this, we use the LBJ Tagger,[6] a state-of-the-art named entity recognition (NER) system [Ratinov and Roth 2009]. LBJ relies on the so-called BILOU scheme: the classifiers are trained to recognize the beginning, the inside, the outside and the last tokens of multitoken entities as well as single token (unit-length) entities. It has been shown that this approach outperforms the more widely used BIO scheme [Ratinov and Roth 2009], which recognizes the beginning, the inside and the outside of an entity name only. LBJ assigns one of four types (`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. In this work, we used Orthomatcher from the GATE framework,[7] 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.

3.1. Architecture

SONEX comprises three independent modules (Figure 3): server, entity extractor, and relation extractor. The server and the entity extractor implement the workflow in Figure 2, as follows. The server fires multiple threads for reading blog posts from the corpus, sending such posts to one entity extractor process, and collecting the results from all entity extractors, storing them in a local database of annotated sentences. Each entity extractor fires a number of threads to process the blog posts from the post queue (we usually set the number of threads to match the number of cores in the host machine). Each thread performs the entire workflow of Figure 2 on a single post. Annotated sentences produced by each thread are stored in the sentence queue and

[4]http://commons.apache.org/lang/.
[5]http://lexsrv3.nlm.nih.gov/SPECIALIST/.
[6]http://cogcomp.cs.illinois.edu/page/software.
[7]http://gate.ac.uk/.

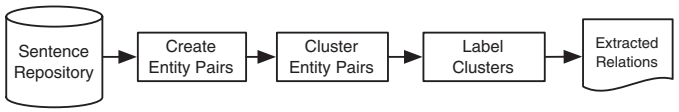


Fig. 4. Relation extraction workflow.

eventually sent back to the server. In our current implementation, we use Berkeley DB[8] as the back-end engine for storing the annotated sentences.

3.2. Postprocessing

Once all sentences are extracted and annotated with entities, we perform a cleaning on the sentences. We found that 20% (around 10 million) of the sentences containing entity pairs were duplicates. We remove exact duplicate sentences using a simple string hashing algorithm based on MD5 [Rivest 1992].

4. EXTRACTING RELATIONS

Figure 4 illustrates the process of relation identification per se, which is done once all annotated sentences are extracted from the corpus. The first step is to build the entity pairs (recall Section 1.1) from the repository of extracted sentences. To accomplish this, we implemented a filtering step that allows us to choose which sentences to be considered for the analysis. For the experiments reported in this article, we used two filtering criteria: (1) the number of words separating the entities in the sentence, which we fix to no longer than 5, as suggested by previous work [Hasegawa et al. 2004]; and (2) the *support* for the entity pair, defined as the number of sentences that contain the entity pair, which we vary in different experiments, as discussed later. Once the sentences are filtered, building the entity pairs consists of extracting the textual features used for clustering, as discussed below.

4.1. Representing Entity Pairs

Following Hasegawa et al. [2004], we use the vector space model (VSM) to represent the context of the entity pairs. That is, we collect the intervening features between a pair of entities for each co-occurrence in the entire dataset, constructing the context vector of the pair. Every pair is represented by a single vector. Our ultimate goal is to cluster entity pairs that belong to the same relation. Regardless of the clustering algorithm in use, the feature space plays an essential role. SONEX can currently use any of the the following features.

—*Unigrams*. The basic feature space containing all stemmed [Allan 1998] single words in the context of a entity pair, excluding stop words.
—*Bigrams*. Many relations may be better described by more than one word (e.g., vice president). For this reason, we include word bigrams, that is, two words that appear in sequence.
—*Part of Speech Patterns (POS)*. Banko and Etzioni claim that many binary relations in English are expressed using a compact set of relation-independent linguistics patterns [Banko and Etzioni 2008]. We assume that a context sentence contains one relation at most. Hence, using the Stanford POS Tagger [Toutanova et al. 2003], we extract one of the predefined part of speech patterns listed in Table II from sentences. If a context sentence contains more than one pattern, only the highest ranked one is extracted. We ranked the patterns according to their frequency on sentences as estimated by previous work [Banko and Etzioni 2008].

[8]http://www.oracle.com/technetwork/database/berkeleydb/.

Table II. Ranked Part of Speech Patterns used by SONEX

Rank	PoS Pattern	Example
1	to+Verb	to acquire
2	Verb+Prep	acquired by
3	Noun+Prep	acquisition of
4	Verb	offered
5	Noun	deal

In building the vectors, we remove all stop words. We consider a feature to be a stop word only if all of its terms appear in the stop words list (e.g., "capital of" is not removed since it contains one term that is not a stop word).

4.2. Clustering Entity Pairs

We use hierarchical agglomerative clustering (HAC) to cluster the entity pairs. HAC is a good option for our task since it does not require the number of clusters in advance. Also, it is used by Hasegawa et al. [2004], and for our task was reported to outperform K-means [Zhang et al. 2005; Rosenfeld and Feldman 2007]. The HAC algorithm starts by placing each entity pair in a distinct cluster and produces a hierarchy of clusters by successively merging clusters with the highest similarity. In our experiments, we cut this hierarchy at a predetermined level of similarity by defining a *clustering threshold*. For example, if the clustering threshold is 0.5, we stop the clustering process when the highest similarity between two clusters is below or equal to 0.5.

To measure the similarity between two clusters, we compared the single, complete, and average link approaches. Single link considers only the similarity between the closest two entity pairs from distinct clusters, while a complete link considers the furthest ones. The average link considers the average similarity between all entity pairs from distinct clusters [Maimon and Rokach 2005; Grossman and Frieder 2004].

4.3. Extracting Relation Names

The last phase is to label every cluster with a descriptive name. Following the state-of-the-art in this area [Treeratpituk and Callan 2006; Glover et al. 2002], SONEX uses information from the cluster itself to extract candidate labels, as follows.

—*Centroid*. The centroid of each cluster (arithmetic mean for each dimension over all the points in the cluster) is computed, and then the feature with the largest mean value is selected as the cluster's label.
—*Standard Deviation (SDEV).* A disadvantage of the centroid method is that the mean can be too biased towards one pair. To mitigate this problem, we propose to penalize terms with a large standard deviation among the cluster's pairs. In this method, the feature to be selected as the label is the one that maximizes the value of the mean divided by its standard deviation among all the pairs within a cluster.

5. WEIGHTING SCHEMES USED IN SONEX

As discussed earlier, the contexts of entity pairs are represented using the vector space model. The state-of-the-art in text clustering assigns weights to the terms according to the standard $tf \cdot idf$ scheme. More precisely, for each term t in the context of an entity pair, tf is the frequency of the term in the context, while

$$idf = \log\left(\frac{|D|}{|d : t \in d|}\right),$$

where $|D|$ is the total number of entity pairs, and $|d : t \in d|$ is the number of entity pairs containing term t. The standard cosine similarity is used to compute the similarity between context vectors during clustering.

Intuitively, the justification for using idf is that a term appearing in many documents (i.e., many contexts in our setting) would not be a good discriminator [Robertson 2004], and thus should weigh proportionally less than other, more rare terms. For the task of relation extraction, however, we are interested specifically in terms that describe relations. Note that in our settings a document is a context vector of one entity pair, which means that the fewer pairs a term appears in, the higher idf score it would have. Consequently, it is not necessarily the case that terms that are associated with high idf weights would be good relation discriminators. On the other hand, popular relational terms that apply to many entity pairs would have relatively lower idf weights. To overcome this limitation, we use a new weight that accounts for the relative discriminative power of a term within a given relation domain, as discussed next.

5.1. The Domain Frequency

It is natural to expect that the relations extracted in SONEX are strongly correlated with a given context. For instance, marriage is a relation between two persons, and thus belongs to the domain PER–PER. We exploit this observation to boost the weight of relational terms associated with marriage (e.g., "wife," "spouse," etc.) in those clusters where the domain is also PER–PER. We do it by computing a *domain frequency* (df) score for every term. The more dominant a term in a given domain compared to other domains, the higher its df score would be.

We start with a motivating example before diving into the details about how we compute domain frequency. We initially built SONEX with the traditional $tf \cdot idf$ and were unsatisfied with the results. Consequently, we examined the data to find a better way to score terms and filter noise. For example, we noticed that the pair Youtube[ORG] – Google[ORG] (associated with the "Acquired by" relation) was not clustered correctly. In Table III we listed all the Unigram features we extracted for the pair from the entire collection sorted by their domain frequency score for ORG–ORG (recall that these are the intervening features between the pair for each co-occurrence in the entire dataset). For clarity, the terms were not stemmed.

Clearly, most terms are irrelevant, which make it difficult to cluster the pair correctly. We listed in bold all terms that we think are useful. Besides "belongs," all these terms have high domain frequency scores. However, most of these terms do not have high idf scores. Term frequencies within a pair are also not helpful in many cases since many pairs are mentioned only a few times in the text. Next, we define the domain frequency score.

Definition. Let P be the set of entity pairs, let T be the set of all entity types, and let $D = T \times T$ be the set of all possible relation domains. The *domain frequency* (df) of a term t, appearing in the context of some entity pair in P, in a given relation domain $i \in D$, denoted $df_i(t)$, is defined as

$$df_i(t) = \frac{f_i(t)}{\sum_{1 \le j \le n} f_j(t)},$$

where $f_i(t)$ is the frequency with which term t appears in the context of entity pairs of domain $i \in D$, and n is the number of domains in D.

Specificity of the df. Unlike the idf score, which is a global measure of the discriminating power of a term, the df score is domain-specific. Thus, intuitively, the df score

Table III. Unigram Features for the pair *Youtube[ORG] – Google[ORG]* with IDF and DF (ORG–ORG) Scores

Term	IDF	DF (ORG–ORG)	Term	IDF	DF (ORG–ORG)
ubiquitious	11.6	1.00	blogs	6.4	0.14
sale	5.9	0.80	services	5.9	0.13
parent	6.8	0.78	instead	4.0	0.12
uploader	10.5	0.66	free	5.0	0.12
purchase	6.3	0.62	similar	5.7	0.12
add	6.1	0.33	recently	4.2	0.12
traffic	7.0	0.55	disappointing	8.2	0.12
downloader	10.9	0.50	dominate	6.4	0.11
dailymotion	9.5	0.50	hosted	5.6	0.10
bought	5.2	0.49	hmmm	9.3	0.10
buying	5.8	0.47	giant	5.4	<0.1
integrated	7.3	0.44	various	5.7	<0.1
partnership	6.7	0.42	revealed	5.2	<0.1
pipped	8.9	0.37	experiencing	7.7	<0.1
embedded	7.6	0.36	fifth	6.5	<0.1
add	6.1	0.33	implication	8.5	<0.1
acquired	5.6	0.33	owner	6.0	<0.1
channel	6.3	0.28	corporate	6.4	<0.1
web	5.8	0.26	comments	5.2	<0.1
video	4.9	0.24	according	4.5	<0.1
sellout	9.2	0.23	resources	6.9	<0.1
revenues	8.6	0.21	grounds	7.8	<0.1
account	6.0	0.18	poked	6.9	<0.1
evading	9.8	0.16	**belongs**	6.2	<0.1
eclipsed	7.8	0.16	authors	7.4	<0.1
company	4.7	0.15	hooked	7.1	<0.1

would favor specific relational terms (e.g., "wife," which is specific to personal relations) as opposed to generic ones (e.g., "member of" which applies to several domains). To validate this hypothesis, we computed the df scores of several relational terms found in the clusters produced by SONEX on the main Spinn3r corpus (details in the next section).

Figure 5 shows the relative df scores of eight relational terms (mayor, wife, CEO, acquire, capital, headquarters, coach, and author) which illustrate well the strengths of the df score. We can see that for the majority of terms (Figure 5(a)–(f)), there is a single domain for which the term has a clearly dominant df score: LOC–PER for mayor, PER–PER for wife, ORG–PER for CEO, and so on.

Dependency on NER Types. Looking again at Figure 5, there are two cases in which the df score does not seem to discriminate a reasonable domain. For coach, the dominant domain is LOC–PER, which can be explained by the common use of the city (or state) name as a proxy for a team as in the sentence "Syracuse football coach Greg Robinson". Note, however, that the problem in this case is the difficulty for the NER to determine that "Syracuse" refers to the university. These are some examples of correctly identified pairs in the coach relation, but in which the NER types are misleading:

—LOC–PER domain: (England, Fabio Capello); (Croatia, Slaven Bilic); (Sunderland, Roy Keane).
—MISC–PER domain: (Titans, Jeff Fisher); (Jets, Eric Mangini); (Texans, Gary Kubiak).

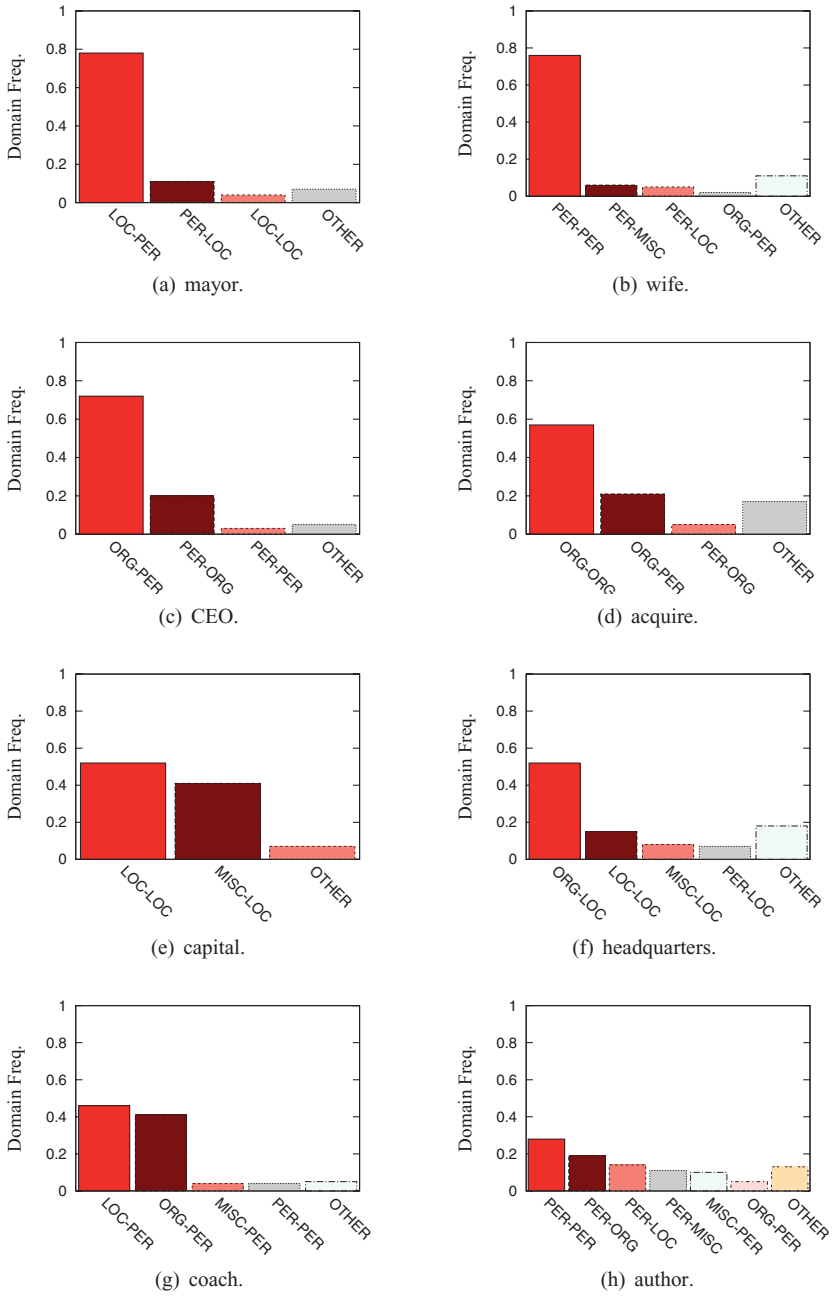


Fig. 5. Domain frequency examples.

This problem is further compounded for the case of author, as book titles (or a part of them) are often proper names of places, persons, organizations, and other kinds of entities, making the task of type identification extremely difficult. Some examples from our experiments are the following:

—PER–PER domain: (Eoin Colfer, Artemis Fowl); (J.K. Rowling, Harry Potter);
—PER–ORG domain: (Donna Cutting, The Celebrity Experience); (Adam Smith, The Wealth of Nations);
—PER–LOC domain: (George Orwell, Animal Farm); (Cormac Mccarthy, The Road).

5.2. Using the df Score

We use df score for two purposes in our work. First, for clustering, we compute the weights of the terms inside all vectors using the product $tf \cdot idf \cdot df$. Second, we also use the df score as a filtering tool by removing terms from vectors whenever their df scores lower than a threshold. Going back to the Youtube[ORG] – Google[ORG] example in Table III, we can see that minimum df filtering helps with removing many noisy terms. We also use maximum IDF filtering, which helps with removing terms that have high df scores only because they are rare and appear only within one domain (e.g., "ubiquitious" (misspelled in source) and "uploader" in this example).

As we shall see in the experimental evaluation, even in the presence of incorrect type assignments made by the NER tool, the use of df scores improves the accuracy of SONEX. It is also worth mentioning that computing the df scores can be done fairly efficiently, and as soon as all entity pairs are extracted.

6. SETUP OF EXPERIMENTAL EVALUATION

Evaluating OIE systems is a difficult problem, especially at the scale with which we employ SONEX, namely, the blogosphere. To the best of our knowledge, no public benchmark exists for the task of information extraction from informal text, such as those often found in social media sites. Furthermore, existing relation extraction benchmarks, such as ACE RDC 2003 and 2004 (recall Section 2), are built from news corpora, whose texts are produced and revised by professional writers and journalists, and, clearly, do not represent the challenges of the task on blogs. Given the lack of benchmarks, the current evaluation approaches rely on manual evaluations (e.g., [Hasegawa et al. 2004; Rosenfeld and Feldman 2007]), whose main limitation is that they do not scale. In fact, it is not even clear whether a manual evaluation through crowd-sourcing (e.g., using Mechanical Turk) would be feasible given that OIE systems, such as SONEX, extract hundreds of thousands of relationships from millions of blog posts.

A different approach to evaluating an information extraction system is to rely on an existing database of facts as the ground truth [Jurafsky and Martin 2009]. This approach, often employed in constrained information extraction settings usually focusing on a specific domain, has the main advantage that it allows for an automatic (and objective) evaluation. However, one disadvantage of this approach is that precision and recall must always be evaluated against the facts that lie in the intersection between the corpus and the reference database.

Our Approach. We combine the two methods above in our work. We build a *reference dataset* for automatic evaluation by automatically matching entity pairs in our clustering task against a publicly available curated database. We call the resulting dataset INTER (for intersection) in the remainder of the article. From INTER, we derive a clean ground truth against which we verify by hand. We build a larger dataset by adding approximately 30,000 entity pairs from our original set into INTER, to study the accuracy of our system in a more realistic scenario. We call this second database, NOISY.

The 30,000 entity pairs in NOISY represent approximately 30% of the total number of extracted entity pairs. We initially created five different samples representing 10%, 20%, 30%, 40%, and 50% of all extracted entity pairs. We got significantly more features

with 30% than with 10% and 20%, but only a few more with 40% and 50%. In addition, we observed that the results for 40% and 50% are similar to those for 30%. Hence, our NOISY dataset is likely to be a representative sample of all entity pairs while requiring significantly less processing time.

We evaluate SONEX by reporting precision, recall, and f-measure numbers for our system running on INTER and NOISY against the ground truth, in a variety of settings. We also report the manual evaluation (conducted by volunteers) of samples of the relationships identified by SONEX, but which are outside of the ground truth. Finally, we report the semantic similarity between the labels identified by SONEX and those in the ground truth.

6.1. Building the Ground Truth

We build the ground truth by automatically matching the entity pairs in our task against a publicly-available, curated database. For the results reported, we used Freebase,[9] a collaborative online database maintained by an active community of users. At the time of writing, Freebase contained over 12 million interconnected topics, most of which correspond to entities in our terminology. Entities in Freebase are connected through properties, which correspond to relationships. For example, "Microsoft" is connected to "Bill Gates" through the property "founders".

Choosing Relations for the Ground Truth. To identify which relations were described by the Spinn3r dataset, we picked three samples of 1,000 entity pairs each. The first sample contains pairs whose support is greater than 300 sentences; the second contains pairs whose support is between 100 and 300 sentences, while the third sample contains pairs whose support is between 10 and 100 sentences. We matched[10] every entity in this sample against the topics in Freebase. Our ground truth then consists of those pairs of topics from Freebase that match entities in our sample and are connected both in Freebase (through a property) and in our sample (by forming an entity pair). We clean the resulting set of entity pairs by standardizing the NER types for all entities that are automatically extracted, hence having all relations homogenous as a result.

Table IV shows the relations in our ground truth and their respective domains and cardinalities.

6.2. Discussion

As outlined above, we test SONEX on two datasets: INTER, which consists of the pairs in the intersection between Freebase and entity pairs extracted from the Spinn3r corpus, and NOISY, which consists of INTER augmented with approximately 30,000 more entity pairs derived from Spinn3r.

The INTER dataset poses, in many ways, similar challenges to those used in the state-of-the-art in OIE for evaluation purposes. Two significant differences are that INTER contains many more relations than in other work that relies on manual evaluation (e.g., Hasegawa et al. [2004] use only two relations), and that INTER contains many entity pairs whose support is lower than the minimum support used in previous work. Both Hasegawa et al. [2004] and Rosenfeld and Feldman [2007] set the minimum support for clustering at 30 sentences, with the justification that this yields better results. (We confirm this observation experimentally in Section 7.4). Instead of 30, we set the minimum support for entity pairs in INTER at 10 sentences. Table V shows a cumulative distribution of the number of pairs for various levels of support in INTER.

[9]http://www.freebase.com.
[10]Using exact string matching.

Table IV. Relations in the Ground Truth. (The column **Freebase Types** shows the types assigned by Freebase, while the column **Domain** shows the closest types that can be inferred by our NER system)

Relation	Freebase Types	Domain	# Pairs
Capital	Country – City/Town	LOC–LOC	77
Governor	State – Governor	LOC–PER	66
Marriage	Person Name – Person Name	PER–PER	42
Athlete Representing	Olym. athlete – Country	PER–LOC	40
Written work	Author – Work written	PER–MISC	26
Headquarters	Company – City/Town	ORG–LOC	21
President	Country – President	LOC–PER	20
Prime Minister	Country – Prime Minister	LOC–PER	18
City Mayor	City/Town – Mayor	LOC–PER	15
Company Founded	Company Founder – Company	ORG–PER	12
Acquired by	Company – Company	ORG–ORG	11
Films Produced	Film Producer – Film	PER–MISC	11
House Speaker	US House of Represent. – Speaker	ORG–PER	7
Album by	Musical Artist – Musical Album	PER–MISC	6
Single by (song)	Musical Artist – Musical Track	PER–MISC	6
Football Head Coach	Football Head Coach – Footb. Team	ORG–PER	5
Products	Company – Product	ORG–MISC	4
Basketball Coach	Basketball Coach – Basket. Team	ORG–PER	3
Vice President	Country – Vice President	LOC–PER	3
Bishop	City/Town – Bishop	LOC–PER	2
	Total		395

Table V. Cumulative Distribution of Number of Pairs (as a function of support for the INTER dataset)

Support Level	Number of Pairs
$\geq$10	395
$\geq$15	300
$\geq$20	247
$\geq$25	214
$\geq$30	176
$\geq$35	147
$\geq$40	133

While INTER reproduces the experimental conditions as in a manual evaluation, it is hardly a representative of the realistic conditions that would be faced by any practical OIE system designed for the blogosphere. We design NOISY with the intent of testing SONEX on a more challenging scenario, by adding thousands of entity pairs that make the clustering task much harder, serving, in a sense, as "noise". Others have used the same approach, but at a much smaller scale: Rosenfeld and Feldman [2007] added 800 "noise" pairs into a ground truth of 200 pairs, while we add approximately 30,000 entity pairs into a ground truth of 395 pairs.

It is important to note that by this ground truth we built, we do not attempt to evaluate the absolute true recall/precision. The problem with a true precision/recall evaluation is this: we can only find the intersection of what the system produces and what is in the reference database (Freebase in our case), but this does not give true precision (as there are many correct extractions that are not in Freebase), nor true recall (as there are facts in the database which are not in the corpus, and hence could

never be extracted in the first place). For recall, the problem is particularly worse because it does not matter how much of Freebase is extracted by the system, what really matters is how much of Freebase is actually in the corpus. To find that out, however, we need a perfect extraction system, as one cannot build a gold-standard manually on 25M blog posts. We are confident in saying that if we did build such a true recall evaluation set from Freebase, the recall of the output of any Open IE system on Spinn3r would be extremely low, as the Spinn3r data were crawled during a two-month period.

6.3. Metrics

We measure the similarity between the *relations* extracted by SONEX and the relations defined in our ground truth, using precision, recall, and their harmonic mean, the f(1)-measure [Manning et al. 2008]. When evaluating clusters, high precision is achieved when most pairs that are clustered together by the OIE system do indeed belong to the same relation in the ground truth. Conversely, high recall occurs when most of the pairs that belong to the same relation in the ground truth are clustered together. As customary, we interpret the f-measure as a proxy for "accuracy" in our discussion.

More precisely, we define two sets S, F containing pairs of entity pairs that belong to the same relation in the output of SONEX and in the ground truth, respectively:

$$S = \{(p,q) \mid p \neq q, \text{ and } p \text{ and } q \text{ are clustered together by SONEX}\}$$
$$F = \{(p,q) \mid p \neq q, \text{ and } p \text{ and } q \text{ belong to the same relation in the ground truth}\}$$

With these, we define

$$\text{precision} = \frac{|S \cap F|}{|S|}, \quad \text{recall} = \frac{|S \cap F|}{|F|}, \quad \text{and} \quad \text{f-measure} = \frac{2 \cdot P \cdot R}{P + R}.$$

7. RESULTS ON THE INTER DATASET

We now report the results on INTER. The first experiment we performed concerned identifying the best settings of the clustering algorithm, as well as the best textual features for clustering. The second experiment studied the impact of pruning terms from the contexts according to their weights (using idf and df scores).

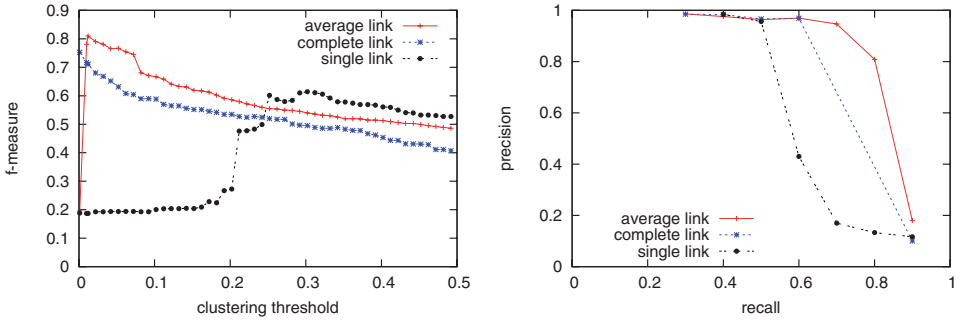
Terminology. For clarity, we will refer to the clustering threshold (recall Section 4.2) as τ in the sequel.

7.1. Comparison of Clustering Methods

Figure 6(a) shows the quality of the clusters produced by three different clustering approaches: single, complete, and average link for $0 \leq \tau \leq 0.5$ (we omit results for $\tau > 0.5$, as the accuracy consistently decreased for all methods in this scenario). In these tests, we use unigrams to build the vectors, and $tf \cdot idf$ as the weighting scheme. The Cosine similarity is used throughout all the experiments.

The behavior of the single link approach is as follows. For $0 \leq \tau \leq 0.2$, this approach yields few relations but with many pairs in them, resulting in high recall but low precision. When $\tau \approx 0.3$, we observed a large improvement in precision at the expense of recall, yielding the best f-measure for this method. However, for $\tau > 0.3$, the decrease in recall is more significant than the increase in precision; consequently, the f-measure value drops.

The behavior of both the average and complete link is much easier to characterize. Complete link yields fairly high precision at the expense of recall even for very small values of τ; further, recall drops consistently as τ increases, without any noticeable increase in precision. Average link truly serves as a compromise between the two other



(a) Clustering methods. (b) Precision vs recall.

Fig. 6. Comparison of clustering methods on INTER (features: unigrams, weights: $tf \cdot idf$).

Table VI. Results for Single, Complete and Average Link Method (when using the best threshold for each of them)

Method	P	R	F1	τ
Single link	0.96	0.46	0.61	0.3
Complete link	0.97	0.62	0.75	0.001
Average link	0.80	0.82	0.81	0.012

methods. As with single link, we observed precision increasing with τ grows, but the best f-score is achieved with a much smaller threshold ($\tau \approx 0.01$). We also noticed a drop in recall as τ grows for average link; however, this drop is not nearly as severe as with the complete link.

Figure 6(b) sheds more light on how each method trades precision and recall. The graph shows the maximum precision for different levels of recall (ranging from 0 to 1 in 0.1 intervals). We observe that all three methods present high precision for recall below 0.45. For single link, the precision quickly decreases for recall values approaching 0.5, while complete and average link still maintain high precision for these recall values. However, average link is able to maintain better recall for higher precision levels than complete link. Recall that values above 0.9 are only achieved when all pairs are grouped into a few big clusters, which leads to poor precision values below 0.2; this is common to all methods.

Table VI shows the best results of each method in our first experiment. Overall, the highest accuracy of all methods is achieved by average link (0.81), outperforming both complete link (0.75) and single link (0.61). For this reason, we used average link as the clustering method for all other experiments we conducted. It is worth mentioning that while we show only the results obtained with unigrams as the clustering feature, we observed the same behavior with the other features as well.

7.2. Comparison of Clustering Features

Figure 7 shows the performance of the different features implemented by SONEX (recall Section 4.1) when using the standard $tf \cdot idf$ weighting scheme compared to $tf \cdot idf \cdot df$.

Several observations are possible from this graph. First, all features performed well, except for bigrams alone. Second, the combination unigrams+bigrams performs the best overall, both when $tf \cdot idf$ alone is used (f-score of 0.82), as well as when df is also used (f-score of 0.87). The part of speech (POS) feature is slightly outperformed by the combination unigrams+bigrams (which, as a matter of fact, subsumed the POS features

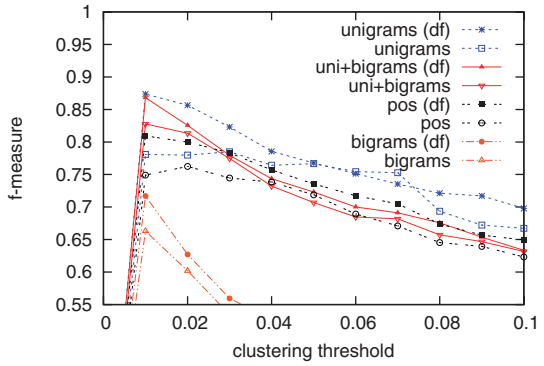


Fig. 7. Comparison of textual features on INTER, $tf \cdot idf$ Vs. $tf \cdot idf \cdot df$ (clustering method: average link).

in our tests). Finally, the use of df increases the f-measure with all features, sometimes substantially (the highest increase was close to 12% in the case of unigrams).

A closer analysis on the impact of the df score revealed that it helps most in cases when a given pair's context vector includes proportionally more nonrelational terms with high idf as opposed to actual relational terms, thus confirming our intuition for the usefulness of this weight. In general, the majority of mis-clustered pairs were those whose contexts contained little useful information about their relationship. For example, the most useful text connecting the pair (⟨AARP, ORG⟩, ⟨Washington, LOC⟩), belonging to the headquarters relation in the ground truth, were "convention in," "gathering this morning in," "poverty event in," and "conference in". Worse still, some entity pairs do not have any context once we remove stop words. Finally, one reason for the poor results produced by using bigrams in isolation is low recall, caused by its inability to extract relations from entity pairs with only one word among them (e.g., "⟨Delaware, LOC⟩ senator ⟨Joe Biden, PER⟩").

It is interesting that the POS feature performed lower than Unigram+Bigrams. The results show that while the best run obtained high precision (0.92), its recall value is significantly lower than the best results achieved by the Unigram+Bigrams feature (0.64 Vs. 0.80). Low recall is expected in rule-based systems. Consider for example the sentence:

"Carmen Electra and ex Dave Navarro were. . . ",

which the Stanford POS tagged as:

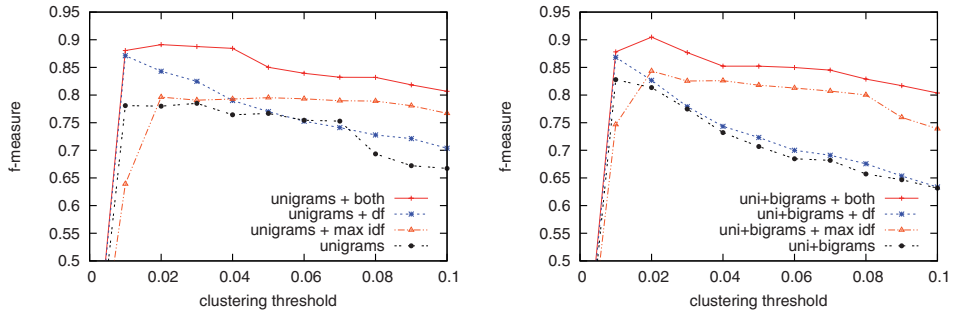
"Carmen/NNP Electra/NNP and/CC ex/FW Dave/NNP Navarro/NNP were/VBD".

The term *"ex"* is important for this pair, but the tagger tags it as a foreign word and we cannot extract it using our POS patterns. If we could train the Stanford tagger on large annotated Web text, we maybe could have improved its accuracy on the Spinn3r collection. However, even then, there are other issues such as sparsity and variability of the relations. For example, we extracted the pair (Eamon Sullivan[PER], Australia[LOC]) from Freebase. This pair belongs to the " Athlete Representing" relation (Olym. athlete Country). This is a difficult relation since many pairs do not include the explicit relation such as " PER representing LOC". Consider for example:

"Eamon Sullivan takes silver for Australia",

tagged as

"Eamon/NNP Sullivan/NNP takes/VBZ silver/NN for/IN Australia/NNP".



(a) Feature: unigrams. (b) Feature: uni+bigrams.

Fig. 8. Comparison of weight-based pruning on different features on INTER using average link

Here "takes silver for" is the only context we extracted for this pair, and the extracted POS pattern is "silver for". This will not match the POS pattern extracted from the pair (Shawn Johnson[PER], US[LOC]) in the sentence "Shawn Johnson won silver putting the US".

Since the best features were unigrams in isolation and the combination unigrams+bigrams, we only use them in our last experiment with INTER.

7.3. Effectiveness of Pruning by Weight

Now, we show the effect of pruning terms from the contexts of entity pairs according to their idf and df scores. We use a *maximum* idf threshold to filter out terms that appear in the context of too few entity pairs. Conversely, we use a minimum df threshold to prune terms within a given context only (in other words, a term with low df score on a given domain may still be used in the context of another entity pair, from a different domain). We experimented with each filter in isolation, and found empirically that the best results were achieved when the maximum idf threshold was set to 5, and the minimum df threshold was set to 0.4.

Figures 8(a) shows the effects of each pruning criterion in isolation, and in combination for when using unigrams only. A similar plot for unigrams+bigrams is shown in Figure 8(b). A general trend is clear in the two scenarios: both pruning methods improve accuracy in a wide range of values for τ. The best f-score (0.89 for unigrams, and 0.90 for unigrams+bigrams) is achieved when both pruning strategies are used. Table VII shows the best overall results on INTER we achieved.

One important issue to consider is that aggressive pruning can have a negative effect on recall. Figure 9 shows the best version of the system with Vs. without feature pruning (refer to Figure 8(b) "uni+bigrams + both" vs. "uni+bigrams"). There is a negative effect on recall in threshold zero (pruning makes a few extra pairs "empty" of context), but we can see that recall drops faster without pruning. Interestingly, we see that pruning can also have a positive effect on recall; not only precision. The main reason is that noisy features increase the number of candidate clusters a pair can be merged with, which makes low precision clusters, but also increases the number of clusters that contain a specific relation; this has a negative effect on recall, as recall is maximized when all the pairs belonging to a specific relation are in the same cluster.

7.4. Effectiveness of Pruning by Support

We also assessed the impact of pruning pairs by their support for the accuracy of the results, motivated by the fact that, in a sense, the support of a pair (i.e., the number of

Table VII. Summary of Results on INTER

Feature	Clustering Method	Max idf	Min df	f-score
Unigrams	avg. link; $\tau = 0.02$	5	—	0.79
	avg. link; $\tau = 0.01$	—	0.4	0.87
	avg. link; $\tau = 0.02$	5	0.4	0.89
Uni+Bigrams	avg. link; $\tau = 0.02$	5	—	0.84
	avg. link; $\tau = 0.01$	—	0.4	0.86
	avg. link; $\tau = 0.02$	5	0.4	0.90

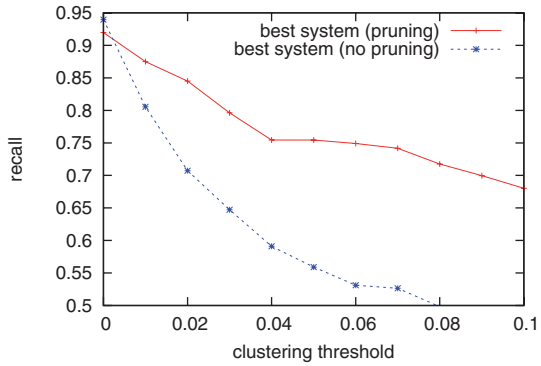


Fig. 9. Recall: with vs. without feature pruning.

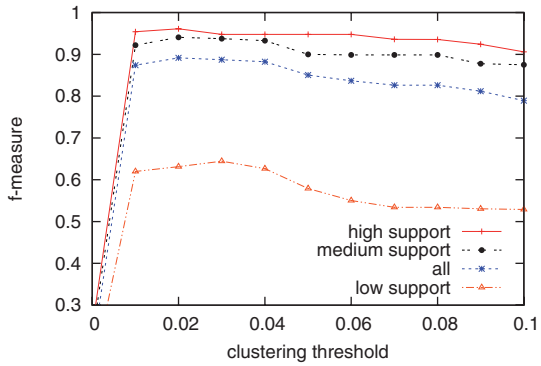


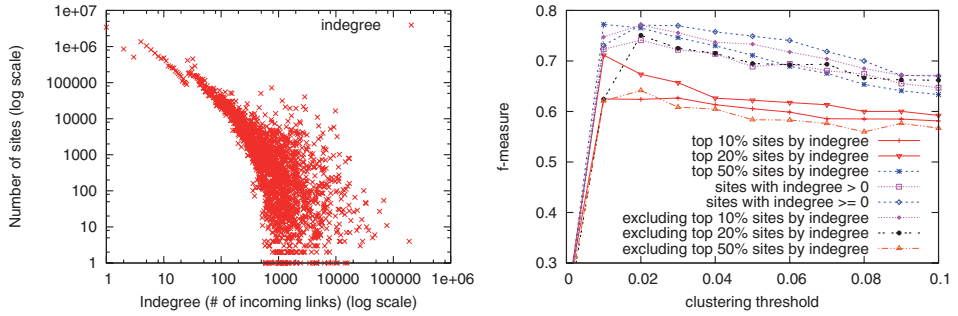
Fig. 10. Effect of support on accuracy in INTER using average link, with unigrams and pruning by df and idf.

sentences for that pair) can be used as crude measure of "popularity" for that pair. We partitioned our evaluation pairs into three groups, according to their support:

—*high*, with support greater than 39 sentences;
—*medium*, with support between 18 and 39 sentences; and
—*low*, with support less than 18 sentences.

This partition yields three roughly equisized subsets: high has 133 pairs, medium has 132 pairs, and low has 130 pairs.

Figure 10 shows the accuracy levels for the different partitions in INTER. (To facilitate the reading of the results, accuracy, in this experiment, is measured relative to the pairs in the given partition.) The graph shows a direct correlation between support



(a) distribution. (b) performance.

Fig. 11. In-degree: The number of incoming links.

and accuracy. This is expected, since the higher the support, the higher the number of terms in the contexts (which, in turn, allows for better clustering). There is a marked difference in accuracy levels when contrasting low-support with medium-support, and medium-support with high-support, indicating that the relationship between support and accuracy is not linear.

7.5. Pruning by In-Degree

Another interesting feature to look at in blogs, which is part of the metadata of the Spinn3r collection and related to the previous experiment on support, is the effect of the number of in-coming links on performance. Figure 11(a) shows the in-degree distribution of the sites in the Spinn3r collection. We can see that the majority of the sites have a low in-degree (there are 8, 989, 885 sites with in-degree of zero). It is worth mentioning that in this experiment we excluded sites that were assigned an in-degree of "−1" in the collection (i.e., unknown). There are two interesting questions to ask.

(1) Do the more "popular" blogs provide more reliable content in a way that performance (accuracy or speed) can be improved?
(2) Sites with extremely high in-degree can be sometimes spam; can we improve SONEX performance by excluding such sites?

Figure 11(b) shows the performance of SONEX (INTER, unigrams, no pruning) on different subsets of the blogs based on in-degree values. The top 10% and top 20% of sites contain most of the ground truth pairs, but not all of them; to avoid low recall due to missing pairs, we removed such pairs from the ground truth of these two subsets. The results show that top 10% and top 20% achieved the lowest scores among the subsets. This bolsters the results from Figure 10 that there is a correlation between "support" of a pair and performance. Interestingly, the run on top 50% achieved only slightly lower results than the best run. This shows that the top 50% provides enough support for successfully cluster the pairs, which is useful since this subset is significantly smaller than the entire collection, and processing it is significantly faster. We also see the effect of excluding the top sites by indegree. Excluding the top 10% and top 20% does not affect the performance; however, excluding the top 50% sites hurts the performance quite a bit, which is not surprising given the results achieved by using only the top 50% subset.

7.6. Summary of Observations on INTER

We can summarize our findings on the experiments on INTER as follows.

—*Clustering Method*
  —Average link consistently outperformed single and complete link in terms of f-measure, as shown in Figure 6(a).
  —Complete link produced the fewest incorrect relationships (i.e., achieved highest precision), as shown in Table VI.
—*Features*
  —The best clustering features are unigrams and the combination of unigrams+bigrams, as shown in Figure 7.
—*Weighting*
  —Using $tf \cdot idf \cdot df$ (instead of $tf \cdot idf$ alone) increased accuracy up to 12%, as shown in Figure 7.
—*Pruning*
  —Pruning terms by maximum idf and minimum df improves accuracy substantially, as shown in Figures 8(a) and 8(b);
—*Results*
  —F-measure values as high as 0.9 were achieved using average link on the combination unigrams+bigrams, with $\tau \approx 0.02$, and $tf \cdot idf \cdot df$, when entity pairs are pruned by $df \geq 0.4$ and $idf \leq 5$, as shown in Figure 8(b). This is an 11% improvement over our baseline setting ($tf \cdot idf$ with unigrams) as proposed by Hasegawa et al. [2004].

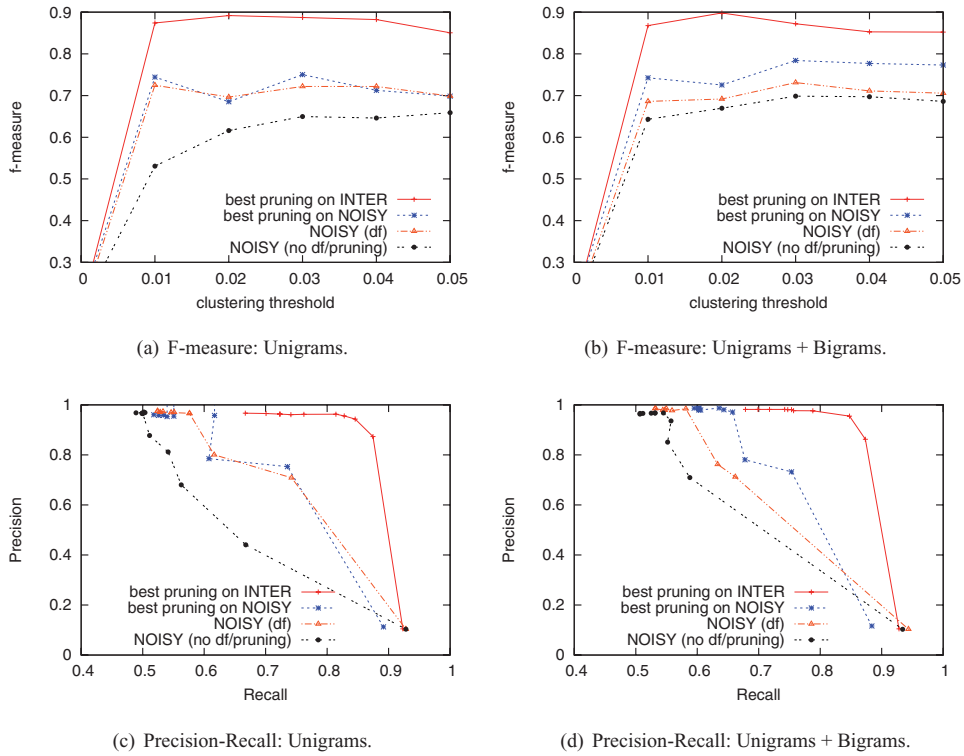
8. RESULTS ON THE NOISY DATASET

We now report our results on NOISY, contrasting them with those on INTER, in order to highlight the challenges of extracting relations in more realistic conditions. First, we show results of an automatic evaluation of SONEX on NOISY, in which, as customary, we perform the clustering on all pairs in NOISY, but report the results on only those pairs known to be in the ground truth. Since average link clustering achieved the best results for NOISY as well, we do not report results for a single and complete link. Next, we complement this analysis with results of a manual evaluation on the pairs not in the ground truth, performed by volunteer computing science students.

8.1. Comparison of Clustering Features

Figure 12 shows a comparison of the accuracy for the unigrams and unigrams+bigrams features on NOISY. The most striking observation is the substantial decrease of the results on NOISY compared to the best results on INTER. We observed a similar drop for the other feature sets as well (POS and bigrams alone). Such a decrease is, of course, expected: NOISY contains not only thousands more entity pairs than INTER, but also hundreds (if not thousands) more relations as well, making the clustering task much harder in practice. Moreover, many context vectors contain terms related to more than one relation because sometimes there is more than one relation between the entities in the pair. Given the approximate nature of text clustering, it is only to be expected that some entity pairs that are clustered together on INTER will be clustered separately on a larger set of entity pairs. In the INTER dataset this is not a problem, since the total number of relations is small, and the pair is likely to end up in a cluster representing its strongest relation.

Another challenge when dealing with NOISY is that it contains, as expected, considerably more NER mistakes, thus affecting the effectiveness of our df score. While on INTER we can expect to find homogeneous relations because we manually corrected all NER-type mistakes for every entity pair, on NOISY, this is virtually hopeless. Now, all domains are inferred from the types assigned by the NER tool—as a result, all df scores decrease. For example, the df score for wife in the PER-PER domain drops from 0.97 on INTER to 0.76 on NOISY. Table VIII lists a comparison between the df scores generated using the INTER and NOISY datasets. Nevertheless, Figure 12 shows that



(a) F-measure: Unigrams.

(b) F-measure: Unigrams + Bigrams.

(c) Precision-Recall: Unigrams.

(d) Precision-Recall: Unigrams + Bigrams.

Fig. 12. Experiments on NOISY.

Table VIII. df Scores for the Dominant Domains (INTER vs NOISY)

dataset	wife	CEO	capital	author	coach	mayor	acquire	HQ
INTER	0.97	0.99	1.00	1.00	0.99	0.99	1.00	0.92
NOISY	0.76	0.92	0.52	0.21	0.41	0.89	0.57	0.52
	PER–PER	ORG–PER	LOC–LOC	PER–MISC	ORG–PER	LOC–PER	ORG–ORG	ORG–LOC

using the minimum df pruning strategy still yields considerably better results than performing no pruning.

8.2. Manual Evaluation of Clusters

We now report on a manual evaluation of SONEX on entity pairs which are not in INTER (nor in the ground truth). Since manual evaluation is expensive and error-prone, we restrict this exercise to samples of the ten largest clusters (each corresponding to each of the ten largest relations in our ground truth; recall Table IV) found by SONEX. The evaluation was performed by eight volunteers. Each volunteer was given a relation name and a list of entity pairs. We report only precision, since recall is unknown, which in this case indicates the fraction of entity pairs in each sample that, in the volunteer's opinion, truly belongs to the given relation. The results shown here correspond to clusters produced using the settings that produced the best results (recall Figure 12).

Table IX shows the precision results obtained in this evaluation. Overall, 75% of the 1098 entity pairs evaluated were deemed correct by the volunteers. However, the precision of individual relations varied greatly (ranging from 0.47 for "Company Founded" to 0.94 for "City Mayor"). The "Company Founded" cluster contains many

Table IX. Manual Evaluation of the Clusters of the 10 Largest Relations

Relation	Domain	Cluster Size (INTER)	Incorrect Types	Precision
Capital	LOC–LOC	182 (64)	29 (16%)	0.87
Governor	LOC–PER	139 (63)	0	0.72
Athlete Representing	PER–LOC	7 (4)	0	0.71
Marriage	PER–PER	129 (13)	1 (0.8%)	0.77
Written work	PER–MISC	105 (17)	60 (57%)	0.84
Headquarters	ORG–LOC	28 (11)	0	0.86
President	LOC–PER	245 (18)	84 (34%)	0.64
Prime Minister	LOC–PER	134 (17)	51 (38%)	0.67
City Mayor	LOC–PER	31 (14)	0	0.94
Company Founded	ORG–PER	98 (9)	9 (9%)	0.47
Total		1098 (230)	234 (21%)	0.75

Company – CEO pairs that do not belong to the "Company Founded" relation; this yields significantly lower results than average. Since there is a large overlap between founders and CEOs in real life (e.g., CEO(Bill Gates, Microsoft) and Founder(Bill Gates, Microsoft)), a cluster containing "Company Founded" pairs would have context related to both founders and CEOs. Therefore, the CEO portion of the context attracts other pairs that belong solely to the CEO relation (e.g., CEO(Eric Schmidt, Google)). Since the HAC algorithm assigns every pair to only one cluster, it would be able to separate both relations with high precision only if every pair in the intersection of "Company Founded" and "Company CEO" is only mentioned in one of the contexts, which is unrealistic to expect.

Two additional observations are worth mentioning. First, the impact of the large number of entity pairs on identifying relations defined by general terms such as "president" was significant. (Further evidence of this issue is given in the "Athlete Representing" cluster in Table X, explained in the next section.) Second, the fraction of entity pairs where at least one entity is assigned an incorrect type from the NER tool is disproportionately higher for domains involving type LOC[11] (4th column in Table IX). The majority of the mistakes are LOC entities labeled MISC.

8.3. Summary of Observations on NOISY

The following general observations can be made from our analysis.

—*Clustering Method*
  —As with INTER, the best clustering method was the average link.
—*Features*
  —As with INTER, the best clustering features were unigrams and unigrams+bigrams.
—*Weighting*
  —The df score improved the results, despite the high number of incorrect domains due to NER mistakes, as shown in Table VIII.
—*Pruning*
  —Pruning terms by maximum idf and minimum df improves accuracy substantially, as shown in Figures 12(a) and 12(b).
—*Results*
  —F-measure values as high as 0.79 were achieved using average link on the combination unigrams+bigrams, using pruning by df and idf, as shown in Figure 12.

[11]A close look at Table IX also shows a large number of errors for the PER–MISC, but this is not surprising, as the MISC type by definition contains many kinds of entities.

Table X. Top Stemmed Terms for the 4 Biggest Relations (represented by the biggest cluster extracted for each relation)

Feature	First (weight)	Second (weight)	Third (weight)
Capital cluster			
unigrams	capit (8.0)	citi (4.7)	coast (2.5)
unigrams+bigrams	capit (14.0)	citi (8.1)	citi of (8.0)
bigrams	s capit (13.2)	capit of (12.8)	citi of (12.7)
POS	capit (4.1)	citi of (3.4)	capit of (3.3)
Governor cluster			
unigrams	gov (220.1)	governor (118.0)	govenor (0.8)
unigrams+bigrams	gov (287.3)	governor (149.5)	s governor (11.4)
bigrams	s governor (22.4)	attorney gener (15.7)	s gov (7.1)
POS	gov (95.1)	governor (62.8)	sen (23.4)
Marriage cluster			
unigrams	wife (37.9)	husband (25.3)	hubbi (6.3)
unigrams+bigrams	wife (30.2)	husband (20.0)	hi wife (1.4)
bigrams	s ex (12.8)	father of (4.9)	the father (4.1)
POS	husband (26.5)	boyfriend (8.7)	wife of (6.3)
Athlete Representing cluster			
unigrams	rep (17.5)	convent (17.4)	congressman (5.6)
unigrams+bigrams	secret (2.6)	met (2.41)	met with (1.0)
bigrams	ambassador as (2.6)	the righteous (2.3)	left of (2.3)
POS	left of (2.6)	winningest (2.1)	decor (1.3)

—The incorrect type identification by the NER tool is disproportionately higher for locations, as shown in Table IX;

—*Manual Evaluation of Clusters*

—Precision levels around 75% on average and as high as 94% were achieved, as shown in Table IX.

It is worth mentioning that SONEX significantly improved over our baseline settings ($tf \cdot idf$ with unigrams) on both INTER and NOISY.

9. EVALUATION OF RELATION NAMES

This experiment aims at evaluating the quality of the labels assigned to the clusters by SONEX. As discussed in Section 4.2, we use two methods for extracting labels from the contexts of the pairs in the resulting clusters: using the *centroid* term, as in the state-of-the-art, and a variant that smoothes out term weights to avoid outliers, which we call SDEV.

Both the Centroid and SDEV methods select one *stemmed term* from the context of one or more entity pairs in the cluster, and, as such, do not always produce a meaningful label. As an illustration, Table X shows the top-3 stemmed terms and their mean weights within a cluster for the four biggest relations on NOISY, across all feature sets we considered. We obtain a more readable (unstemmed) label for each cluster based on the frequency of original terms corresponding to a given stem. For example, in our experiments, "capit" becomes "capital" since this is the most frequent term among all terms that stem to "capit".

Further Evidence of the Difficulty of Extracting General Relations. In passing, we point to Table X to return to the difficulty of automatically extracting the "Athlete Representing" relation, connecting athletes and their countries,[12] whose main relational

[12]This was a popular topic at the time the Spinn3r data was collected, right after the Beijing Olympic games.

Table XI. INTER Labels

Relation	Omiotis			Manual		
	Centroid	SDEV	Difference	Centroid	SDEV	Difference
Capital	5.00	5.00	*0.00*	5.00	4.50	*0.50*
Governor	3.97	2.03	*1.94*	4.42	3.45	*0.97*
Athlete Repr.	1.00	4.76	*3.76*	2.66	4.30	*1.10*
Marriage	3.97	3.97	*0.00*	4.60	4.60	*0.00*
Author	3.92	2.96	*0.96*	4.18	4.89	*0.71*
Headquarters	4.55	1.15	*3.40*	4.47	3.97	*0.50*
President	4.60	4.60	*0.00*	4.52	4.52	*0.00*
Prime Minister	4.89	4.89	*0.00*	3.72	3.72	*0.00*
Mayor	5.00	5.00	*0.00*	4.80	4.80	*0.00*
Founder	5.00	5.00	*0.00*	5.00	4.00	*1.00*
Average	4.19	3.94	*0.25*	4.33	4.27	*0.06*

term is "represent". Because there are many different relations expressed through the verb "represent" besides being an Olympic athlete (e.g., as an elected official), entity pairs from this relation end up clustered within these other relations *and* vice-versa. Evidence of this is the fact that, among the top-3 terms for this cluster, we can find terms indicating political representation ("congressman" and "ambassador") as well as other very generic terms ("met with").

9.1. Evaluation Method

There is still no standard evaluation methodology for cluster labeling, and there are no standard benchmarks to compare alternative labeling methods [Carmel et al. 2009]. To evaluate an extracted label, we resort to the semantic relatedness of the unstemmed term with the relation name corresponding to the relevant cluster in our ground truth. Semantic relatedness methods are widely used in text mining tasks [Gabrilovich and Markovitch 2007]; we used Omiotis,[13] a system for measuring the relatedness between words, based on WordNet[14] [Tsatsaronis et al. 2010]. Omiotis reports relatedness in a scale from 1 to 5, where 1 indicates very weak relatedness and 5 indicates very strong relatedness. In addition, for comparison we repeated a similar evaluation conducted manually and using the same relatedness scale. This manual assessment was done by four computing science students who did not participate in the previous evaluation.

Table XI presents the average relatedness levels for the Centroid and SDEV methods on the INTER dataset, while Table XII shows results for the clusters produced from the NOISY dataset, after removing entity pairs that are not in the intersection. Consistent with the clustering results, the INTER labels are more accurate than the ones obtained on NOISY. Also, on average, Centroid outperforms SDEV on all evaluations. Overall, the relatedness scores reported by Omiotis are very close to the manually assigned ones, with minor differences ranging from 0.14 to 0.33 on average. No labels show high discrepancy between the Omiotis and manual assessment. The only relation with consistent assessment of low score on NOISY is "Athlete Representing". From our observations we learn that this relation is not often mentioned explicitly in the text, which makes it hard to identify using our approach that only looks for clues in text between two entities. Varied results between relations is very common in previous works as well, with systems that use $tf \cdot idf$ only for weighting [Hasegawa et al. 2004].

[13]http://omiotis.hua.gr.

[14]http://wordnet.princeton.edu/.

Table XII. NOISY Labels

Relation	Omiotis			Manual		
	Centroid	SDEV	Difference	Centroid	SDEV	Difference
Capital	4.60	2.91	*1.69*	4.30	2.09	*2.21*
Governor	3.91	2.05	*1.86*	4.17	1.35	*2.82*
Athlete Repr.	1.29	1.88	*0.59*	1.95	2.47	*0.52*
Marriage	3.66	2.90	*0.76*	3.96	4.30	*0.34*
Author	2.86	3.05	*0.19*	3.73	3.75	*0.02*
Headquarters	3.60	3.35	*0.25*	3.20	3.31	*0.11*
President	3.15	2.35	*0.80*	4.58	3.52	*1.06*
Prime Minister	4.56	4.56	*0.00*	3.68	3.70	*0.02*
Mayor	4.73	5.00	*0.27*	4.81	4.81	*0.00*
Founder	3.00	3.00	*0.00*	3.50	3.25	*0.20*
Average	3.54	3.10	*0.64*	3.78	3.25	*0.52*

Table XIII. Annotators Agreement with Kappa Statistics

Annotators	Observed Agreement (By Chance)	Kappa	Weighted Kappa
A1–A2	61.11% (26.92%)	0.46	0.68
A1–A3	53.70% (24.18%)	0.38	0.62
A1–A4	55.56% (23.29%)	0.42	0.63
A2–A3	59.26% (27.91%)	0.43	0.66
A2–A4	50.00% (24.86%)	0.33	0.62
A3–A4	52.94% (24.07%)	0.38	0.65

Table XIV. Label Examples

Relation	INTER		NOISY	
	Centroid	SDEV	Centroid	SDEV
Capital	capital	city	capital	living
Governor	gov	delegation	gov	today
Athlete Representing	won	representing	rep	representative
Marriage	husband	wife	wife	married
Written work	book	wrote	book	book
Headquarters	headquarters	based	headquarters	headquarters
President	president	president	presidential	political
Prime Minister	prime	prime	minister	minister
Mayor	mayor	mayor	mayor	mayor
Founder	founder	cofounder	chairman	head

Table XIII shows the degree of agreement among the annotators using the Kappa statistical measure [Fleiss et al. 2003]. The degree of agreement ranges from 50% to 62%, which is significantly better than the degree of agreement by chance. However, these results demonstrate the challenge of finding labels that satisfy different users. We do not have any instance where none of the annotators agree on, and also no high discrepancy among annotators (e.g., scores 1 and 5 for the same label). Most disagreements differ by a single score. The "Kappa" column only considers exact matches between annotators. On the other hand, the "Weighted Kappa" column considers the difference between scores (e.g., 4 is closer than 3 when compared with 5).

Example Relation Names. Table XIV shows the highest ranked relation names extracted by SONEX on both INTER and NOISY for the ten largest clusters.

10. COMPARISON TO REVERB

In the previous sections, we presented an extensive evaluation of SONEX and how it extends the work of Hasegawa et al [2004] for extracting relations from the blogosphere. In this section, we perform a comparative experiment between our system and ReVerb, a state-of-the-art system [Fader et al. 2011]. We use the implementation distributed by the authors.[15]

One of the challenges of evaluating systems like SONEX and ReVerb is that their outcomes are not the same. ReVerb recognizes relationships at sentence level. For example, ReVerb would extract the relation "is opponent of" from the sentence "Obama is opponent of McCain". Each of these sentence-level extractions is often evaluated as correct or incorrect by human judges. On the other hand, SONEX extracts relations at corpus level. In order to compare both systems at the same level, we convert ReVerb's extractions into corpus-level ones by applying a simple aggregation method proposed by TextRunner [Banko et al. 2007]. All pairs of entities connected through a specific relation (e.g., "is opponent of") in at least a sentence are grouped together into one cluster. Observe that this process may produce *overlapping clusters*, that is, two clusters may share entity pairs.

The evaluation measures used in the previous sections cannot be used to evaluate overlapping clustering. Therefore, we adopt the following measures for this experiment: purity and inverse purity [Amigó et al. 2009]. These measures rely on the precision and recall of a cluster C_i given a relation R_j:

$$\text{precision}(C_i, R_j) = \frac{|C_i \cap R_j|}{|C_i|} \qquad \text{recall}(C_i, R_j) = \text{precision}(R_j, C_i).$$

Purity is computed by taking the weighted average of maximal precision values:

$$\text{purity} = \frac{1}{M} \sum_i \arg\max_j \quad \text{precision}(C_i, R_j),$$

where M is the number of clusters. On the other hand, inverse purity focuses on the cluster with maximum recall for each relation:

$$\text{inverse purity} = \frac{1}{N} \sum_j \arg\max_i \quad \text{recall}(C_i, R_j),$$

where N is the number of relations. A high purity value means that the clusters produced contain very few undesired entity pairs in each cluster. Moreover, a high inverse purity value means that most entity pairs in a relation can be found in a single cluster.

The INTER dataset was used in this experiment. We provide both SONEX and ReVerb with the same sentences and entities. We configured SONEX with the best setting we found in previous experiments: average link with threshold 0.02, the unigrams+bigrams feature set, the $tf \cdot idf \cdot df$ weighting scheme and pruning on idf and df.

Table XV presents the results for SONEX and ReVerb. Observe that both systems achieved very high purity levels, while SONEX shows a large lead in inverse purity. The low inverse purity value for ReVerb is due to its tendency to scatter entity pairs of a relation into small clusters. For example, ReVerb produced clusters such as "is acquired by," "has been bought by," "was purchased by," and "was acquired by," all containing

[15]http://reverb.cs.washingtonx.edu/.

Table XV. Comparison Between SONEX and ReVerb

Systems	Purity	Inv. Purity
ReVerb	**0.97**	0.22
SONEX	0.96	**0.77**

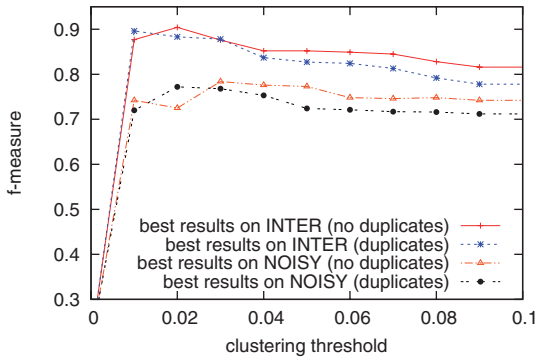


Fig. 13. Results on Inter and NOISY: Duplicates vs. no duplicates.

small subsets of the relation "Acquired by". This phenomenon can be explained by ReVerb's reliance on sentence-level extractions and the variety of ways a relation can be expressed in English. These results show the importance of relation extraction methods that work beyond sentence boundaries, such as SONEX.

11. APPLYING OIE ON THE BLOGOSPHERE

Finally, we highlight the main issues we encountered in extracting relations from the blogosphere and how we address them.

(1) Duplicate content that skewed the true distribution of terms. Figure 13 shows that duplicates indeed hurt performance on both INTER and NOISY; hence, we eliminated duplicates. The problem with duplicates is that they affect the weights assigned to features. The more times a feature appears in a context of a pair, the greater effect it has (the term frequency part). But what we really want to know is the term frequency of the features in different occurrences of the pair (such as different blog posts) rather than counting the same occurrence (same source) multiple times just because the text was duplicated. This also affects the domain frequency and idf weights.
(2) Misspellings and noise in general. We used a filtering on the idf score to get rid of noisy terms (e.g., "ubiquitious" and "hmmm" in Table III)
(3) The performance of the NER tool was shown to be lower for blog posts than for news articles (see [Ratinov and Roth 2009]). The filtering based on domain frequency helped us to get rid of many misclassified or wrong entities (i.e., wrong boundary). For example, the NER created the pair Ted Mcginley[PER] – Children[ORG] from the sentence "Ted Mcginley from Married with Children". Since both the entity type and boundary are wrong, it is better to exclude this pair. The only feature we extracted for this pair is "married". Since the domain frequency for "married" in the domain `PER–ORG` is extremely small, this feature was filtered and consequently the pair was filtered for having no features. Overall, we filtered 3336 pairs (approximately 11% of the NOISY dataset) using the filtering on idf and df in the experiments with the NOISY dataset.

These are all blog related, although not unique only to blogs, and are relevant to the web at large.

12. CONCLUSION

We presented an OIE system for extracting information networks from the blogosphere that works by identifying named entities from the text, clustering the sentences that connect those entities, and extracting prominent terms from the clusters to use as labels. We introduced the domain frequency (df) score, a new term-weighting score designed for identifying relational terms within different domains, and showed that it substantially increases the accuracy of our system in every test we performed. We believe that df can be utilized in various applications, with the advantage that in practice, for many such applications, the list of terms and scores can be used off-the-shelf with no further effort. Also, the df score computation is based on probability (we do not consider the NER to be part of it), and, as such, it can be utilized in other languages with a similar structure to English.

We reported the first results on large-scale experiments with clustering-based OIE systems on social media text, studying the effect of several parameters for such an approach. We also discussed an automatic way of obtaining high-quality test data from an online curated database. Finally, our experimental evaluation showed textual clustering techniques to be a viable option for building an OIE system for the blogosphere. More importantly, our results shed some light on the accuracy of state-the-art extraction tools in this setting, as well as on ways of tuning such systems for higher accuracy.

Given the results obtained in our experiments, there are a few observations we would like to make. First, the HAC method is not capable of extracting more than one relation for a pair, which makes it a poor choice for some pairs. Therefore, we plan to study clustering algorithms that are able to assign an entity pair to multiple clusters. Second, we see that the performance of a relation extraction system is dependent, among others, on the relation sought. To learn the capabilities and shortcomings of a system, an evaluation set needs to include hundreds of different relations. To the best of our knowledge, there is no such resource available. Using the same Freebase method we presented here, we plan to construct a significantly larger evaluation set than any of the existing ones. This would help to overcome the problem of lack of a benchmark for extracting relations from the blogosphere.

Recently, external resources have been successfully utilized to enhance document and cluster labeling [Carmel et al. 2009; Syed et al. 2008]. The main idea is to use the cluster's important terms to find relevant pages in a knowledge base (e.g., Wikipedia), and then extract candidate labels from those pages. This approach has the potential to enrich the labels currently produced by SONEX; we plan further experiment it in the future.

We encourage the reader to visit our website and explore the data and results further.

REFERENCES

AGICHTEIN, E. AND GRAVANO, L. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM Conference on Digital Libraries*. ACM, New York, 85–94.

ALLAN, J. 1998. Book review: Readings in information retrieval edited by K. Sparck Jones and P. Willett. *Inf. Process. Manage. 34*, 4, 489–490.

AMIGO, E., GONZALO, J., ARTILES, J., AND VERDEJO, F. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retrieval 12*, 461–486.

BANKO, M., CAFARELLA, M. J., SODERLAND, S., BROADHEAD, M., AND ETZIONI, O. 2007. Open information extraction from the Web. In *Proceedings of the IJCAI*. M.M. Veloso Ed., 2670–2676.

BANKO, M. AND ETZIONI, O. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics, 28–36.

BONTCHEVA, K., DIMITROV, M.,MAYNARD, D., TABLAN, V., AND CUNNINGHAM, H. 2002. Shallow methods for named entity co-reference resolution. In *Proceedings of TALN*.

BRIN, S. 1998. Extracting patterns and relations from the world wide web. In *Proceedings of WebDB*. 172–183.

BUNESCU, R. C. AND MOONEY, R. J. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT '05)*, R. J. Mooney Ed., Association for Computational Linguistics, Morristown, NJ, 724–731.

BUNESCU, R. C. AND MOONEY, R. J. 2007. Learning to extract relations from the web using minimal supervision. *ACM Trans.Web*. To appear.

BURTON, K., JAVA, A., AND SOBOROFF, I. 2009. The ICWSM 2009 spinn3r dataset. In *Proceedings of the Annual Conference on Weblogs and Social Media*.

CARMEL, D., ROITMAN, H., AND ZWERDLING, N. 2009. Enhancing cluster labeling using wikipedia. In *SIGIR*. 139–146.

CHEN, J., JI, D., TAN, C.L., AND NIU, Z. 2005. Unsupervised feature selection for relation extraction. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*. Springer, Berlin.

CNN. 2008. McCain ad compares Obama to Britney Spears, Paris Hilton. http://www.cnn.com/2008/POLITICS/07/30/mccain.ad.

CRAVEN, M., DIPASQUO, D., FREITAG, D., MCCALLUM, A., MITCHELL, T.M., NIGAM, K., AND SLATTERY, S. 2000. Learning to construct knowledge bases from the world wide web. *Artif. Intell. 118*, 1-2, 69–113.

CULOTTA, A. AND SORENSEN, J.S. 2004. Dependency tree kernels for relation extraction. In *Proceedings of ACL*. 423–429.

DODDINGTON, G., MITCHELL, A., PRZYBOCKI, M., RAMSHAW, L., STRASSEL, S., AND WEISCHEDEL, R. 2004. The automatic content extraction (ace) program–tasks, data, and evaluation. In *Proceedings of LREC*. 837–840.

ETZIONI, O., CAFARELLA, M., DOWNEY, D., KOK, S., POPESCU, A.-M., SHAKED, T., SODERL, S., , WELD, D. S., AND YATES, A. 2004. Web-scale information extraction in Knowitall: (Preliminary results). In *Proceedings of the 13th International Conference on the World Wide Web (WWW '04)*. ACM, New York, 100–110.

FADER, A., SODERLAND, S., AND ETZIONI, O. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*. Association for Computational Linguistics, 1535–1545.

FISHER, D., SODERLAND, S., FENG, F., AND LEHNERT, W. 1995. Description of the UMASS system as used for MUC-6. In *Proceedings of the 6th Conference on Message Understanding (MUC6 '95)*. Association for Computational Linguistics, Morristown, NJ, 127–140.

FLEISS, J. L., LEVIN, B., AND PAIK, M. C. 2003. *Statistical Methods for Rates and Proportions* 3rd Ed., Wiley, New York.

GABRILOVICH, E. AND MARKOVITCH, S. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI '07)*. Morgan Kaufmann, San Francisco, CA, 1606–1611.

GLOVER, E. J., TSIOUTSIOULIKLIS, K., LAWRENCE, S., PENNOCK, D. M., AND FLAKE, G. W. 2002. Using Web structure for classifying and describing Web pages. *In Proceedings of the 11th International Conference on the World Wide Web (WWW '02)*. ACM, New York, 562–569.

GROSSMAN, D. A. AND FRIEDER, O. 2004. *Information Retrieval: Algorithms and Heuristics* 2nd Ed. Springer, Berlin.

GUODONG, Z., JIAN, S., JIE, Z., AND MIN, Z. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL '05)*. Association for Computational Linguistics, Morristown, NJ, 427–434.

HANNEMAN, R. AND RIDDLE, M. 2005. Introduction to social network methods. http://faculty.ucr.edu/òhanneman/nettext/.

HASEGAWA, T., SEKINE, S., AND GRISHMAN, R. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL '04)*. Association for Computational Linguistics, Morristown, NJ, 415.

JURAFSKY, D. AND MARTIN, J. H. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics* 2nd Ed. Prentice-Hall, Englewood Cliffs, NJ.

KAMBHATLA, N. 2004. Combining lexical, syntactic and semantic features with maximum entropy models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04)*.

KNOX, H., SAVAGE, M., AND HARVEY, P. 2006. Social networks and the study of relations: Networks as method, metaphor and form. *Economy Soc. 35*, 1, 113–140.

MAIMON, O. AND ROKACH, L. (EDS.). 2005. *The Data Mining and Knowledge Discovery Handbook*. Springer, Berlin.

MANNING, C. D., RAGHAVAN, P., AND SCHTZE., H. 2008. *Introduction to Information Retrieva*l. Cambridge University Press, Cambridge, UK.

MARLOW, C. 2004. Audience, structure and authority in the weblog community. International Communication Association.

MINKOV, E. AND WANG, R.C. 2005. Extracting personal names from emails: Applying named entity recognition to informal text. In *Proceedings of the HLT-EMNLP*.

MINTZ, M., BILLS, S., SNOW, R., AND JURAFSKY, D. 2009. 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 InternationalJoint Conference on Natural Language Processing of the AFNLP*. Vol. 2, Association for Computational Linguistics, Morristown, NJ, 1003–1011.

RATINOV, L. AND ROTH, D. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL '09)*. Association for Computational Linguistics Morristown, NJ, 147–155.

RIVEST, R. 1992. The MD5 message-digest algorithm. RFC 1321. MIT and RSA Data Security.

ROBERTSON, S. 2004. Understanding inverse document frequency: On theoretical arguments for IDF. *J. Documentation 60*.

ROSARIO, B. AND HEARST, M.A. 2004. Classifying semantic relations in bioscience texts. In *Proceedings of the ACL*. 430–437.

ROSENFELD, B. AND FELDMAN, R. 2007. Clustering for unsupervised relation identification. In *Proceedings of CIKM '07*, ACM, New York, 411–418.

SHINYAMA, Y. AND SEKINE, S. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings on the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics, Morristown, NJ, 304–311.

SYED, Z., FININ, T., AND JOSHI, A. 2008. Wikipedia as an ontology for describing documents. In *Proceedings of the Second International Conference on Weblogs and Social Media*. AAAI Press.

TOUTANOVA, K., KLEIN, D., MANNING, C. D., AND SINGER, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL*.

TREERATPITUK, P. AND CALLAN, J. 2006. Automatically labeling hierarchical clusters. In *Proceedings of the International Conference on Digital Government Research*. ACM, New York, 167–176.

TSATSARONIS, G., VARLAMIS, I., AND VAZIRGIANNIS, M. 2010. Text relatedness based on a word thesaurus. *J. Artif. Intell. Res. 37*, 1–39.

WU, F. AND WELD, D. S. 2010. Open information extraction using Wikipedia. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-10)*.

ZELENKO, D., AONE, C., AND RICHARDELLA, A. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res. 3*, 1083–1106.

ZHANG, M., SU, J., WANG, D., ZHOU, G., AND TAN, C.L. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Proceedings of IJCNLP*. 378–389.

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 18th International Conference on the World Wide Web (WWW '09)*. ACM, New York, 101–110.

Received September 2010; revised January 2012; accepted April 2012