A Bootstrapping Approach for Identifying Stakeholders in Public-Comment Corpora

Jaime Arguello
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
jaime@cs.cmu.edu

Jamie Callan
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
callan@cs.cmu.edu

ABSTRACT

A stakeholder is an individual, group, organization, or community that has an interest or stake in a consensus-building process. The goal of stakeholder identification is identifying stakeholder mentions in natural language text. We present novel work in using a bootstrapping approach for the identification of stakeholders in public comment corpora. We refine the definition of a stakeholder by categorizing stakeholders into 2 distinct stakeholder types and experiment with automatically identifying one of these two types: instances where the author identifies him/herself as a member of a particular group. An existing bootstrapping information extraction algorithm is combined individually with 3 distinct extraction pattern templates. Results show that this stakeholder group can be learned in a minimally supervised bootstrapping framework using 2 of the 3 extraction pattern templates. An experimental analysis explores the challenges in applying the third extraction pattern template to this problem. Results on all 3 extraction pattern templates provide insight on the unique and novel challenge of identifying stakeholders.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Abstracting methods, Dictionaries, Indexing methods, Linguistic processes.

General Terms

Algorithms, Performance, Experimentation.

Keywords

Stakeholder identification, information extraction, eRulemaking, public comments, information retrieval, text analysis.

1. INTRODUCTION

A stakeholder is a person or group of people that have an interest in the outcome of a policy change or consensus-building process. Merriam-Webster[1] defines a stakeholder as "one who has a stake or share in an undertaking or enterprise." The goal of stakeholder identification is locating stakeholder mentions in free, natural language text, in this case in public-comment corpora. The approach described in this paper identifies stakeholder mentions by learning a set of patterns that represent linguistic expressions that faithfully signal a stakeholder mention. Extraction patterns are learned from unlabeled data starting with only a small set of example stakeholders supplied by the user. Although our approach borrows ideas and algorithms from information extraction (IE), to the best of our knowledge, stakeholder identification has not been explicitly addressed in previous work.

Our work in stakeholder identification is done as part of the eRuleMaking project[2]. The goal of the eRulemaking project is to produce novel text-mining applications for U.S. regulatory agencies. U.S. law and standard regulatory practice requires U.S. regulatory agencies to give notice of a proposed rule and then to respond to *substantive* comments from lobbyists, companies, trade organizations, special interest groups, and the general public before issuing a final rule. One necessary step in addressing all substantive comments is discovering which specific individuals, groups, and communities care about or will be affected by a specific regulation. It is a priority for U.S. government employees to address public comments written by or on behalf of (1) people who are directly affected by the proposed regulation and (2) people who have subject matter expertise on the issue at hand. For high-profile regulations that attract hundreds of thousands of public comments, such as the Environmental Protection Agency's (EPA) Mercury Rule (USEPA-OAR-2002-0056), this is a daunting task if done by hand.

We approach the problem of stakeholder identification by adopting a semi-supervised bootstrapping framework. The system takes as input an unannotated corpus of public-comments and a set of seed stakeholders. The seed stakeholders are prototypical stakeholders known beforehand to be frequent and high-quality, meaning that they mostly occur in contexts where they are stated as a stakeholder. Our bootstrapping algorithm is structurally identical to the meta-bootstrapping approach detailed first in [8]. Several extraction pattern templates were evaluated, including the verb-centric extraction pattern templates used in [8]. The best performance in terms of f-measure was obtained using a part-of-speech based extraction pattern template that imposes a semantic constraint on the extracted NP. We evaluate our approach on a 1,020 document test set annotated by a single coder according to a coding scheme. The validity of the single coder's annotations was evaluated in terms of the agreement with respect to a second coder's annotations on 50% of this test set.

[1] http://www.merriam-webster.com

[2] http://erulemaking.ucsur.pitt.edu/

The remainder of the paper is organized as follows. Section 2 motivates and explains the categorization of the stakeholders into two distinct types. Some details about the coding scheme refinement process are described to help the reader understand what is meant by stakeholder in this work. Section 3 surveys relevant work. Section 4 describes the bootstrapping algorithm and the 3 distinct extraction pattern templates individually evaluated under the same bootstrapping framework. Section 5 discusses evaluation and analysis of results. Conclusions and a discussion of future work are presented in Section 6.

2. STAKEHOLDER DEFINITION

A stakeholder is an abstract concept. Before experimenting with approaches for automatically extracting them, it was necessary to more formally define what a stakeholder is. The most general definition of a stakeholder is an entity that has a vested interest in the outcome of a decision-making process. Parting from this general definition, the data was examined to see if different types of stakeholders exist and to survey the different types of contexts that stakeholders are mentioned in. The corpus used was the "Mercury Corpus", a public-comment corpus made available by the Environmental Protection Agency.[3] It is a collection of more than 500,000 public comments written to the EPA in response to their 2004 proposal of new national emissions standards for hazardous air pollutants for coal- and oil-fired, steam-generating, utility plants. The Mercury Corpus was chosen for this round of analyses and experiments because it has been used in previous text-mining research [4] [12] [13] [14] and because of its size. It contains about 120,000 completely original or modified form letter comments. Faced with this much data, automatic stakeholder identification would be useful to U.S. government agencies such as the EPA.

Preliminary analyses of the data revealed that stakeholders can be categorized into 2 distinct types, defined below with accompanying examples taken directly from the Mercury Corpus.

1. **Author Self ID:** The author identifies herself as being a member of a particular group or community. By doing so, the author implies that this community has an interest in the regulation, either because they are or will be directly affected and/or because they possess technical expertise in the matter.
    a. "As a woman of child-bearing age, this concerns me…"
    b. "I am an avid fisherman and I do not agree with…"
    c. "I am a retired scientist who worked in a lab employing mercury…"
2. **Impacted Entity:** The author speaks on behalf of a group. The author mentions that this entity is or will be impacted by the new rule. This may be a group or community of which the author is not a member.
    a. "Mercury pollution can cause […] damage in young children."
    b. "Mercury is especially harmful to pregnant women.

The general stakeholder category was divided into these two more focused categories for two reasons. First, preliminary analyses revealed that these two stakeholder types tend to occur in different contexts. **Author Self ID** mentions tend to occur near the start of the sentence. **Impacted Entity** mentions tend to occur in the object position of verbs such as "affect", "hurt", and "destroy". Thus, it is reasonable that learning the two independently with two mutually exclusive sets of extraction patterns might be more effective than attempting to learn both groups with the single set of extraction patterns. The second reason is that the user community might want to keep these two stakeholder types separate. The user community might want to know if a stakeholder mention is an **Author Self ID** or an **Impacted Entity**.

After defining **Author Self ID** and **Impacted Entity**, a coding scheme was constructed to more narrowly define which noun phrases should and should not qualify as a stakeholder mention. Some highlights of the coding scheme refinement process are detailed below to give the reader a better idea of what is called a stakeholder in this work.

The first iteration of the coding scheme was tested on a set of 45 public comments from the Mercury Corpus. Four independent coders were asked to mark spans of text corresponding to **Author Self ID** and **Impacted Entity**. On these 45 documents, each coder found an average of 18 **Author Self ID** mentions and 153.75 **Impacted Entity** mentions. The Kappa values were 0.48 and 0.30 for **Author Self ID** and **Impacted Entity**, respectively. The Kappa (overlap) values were 0.86 and 0.67. Kappa (overlap) considers two predictions to be equivalent if there is any word-overlap between both spans of text extracted (e.g., "a sufferer of mercury toxicity" and "a sufferer of mercury toxicity and presently going through detoxification"). Regular Kappa considers two predictions equivalent only if there is an exact match between both spans of text. Thus, regular Kappa is at best equal to Kappa (overlap), but is expected to be lower.

An analysis of the stakeholders marked by the coders motivated further refinement of our coding scheme, particularly by expanding the list of stakeholder exclusions. For **Author Self ID**, coders marked spans of text that would require a reader to infer who the stakeholder is (i.e., the stakeholder is not mentioned explicitly in the statement). For example, coders selected spans of text such as "in my first trimester of pregnancy" and "I am fighting cancer". A human reader can easily infer that these two statements imply that "pregnant mothers" and "cancer survivors" are two groups or communities that authors belong to. However, the stakeholder group is not explicitly mentioned in the discourse. Thus, a rule is imposed that a stakeholder mention must be a noun-phrase (NP) that explicitly identifies a stakeholder entity. Furthermore, in the **Impacted Entity** group, coders selected entities that are not people or communities of individuals. Coders selected things such as "our wildlife", "our fish", and "our environment". Although these might have some value to our user community (i.e., it might be valuable to know that mercury pollution contaminates fish, and consequently affects fish eaters and fishermen), we restricted ourselves to individuals, communities, and organizations. Finally, for both **Author Self ID** and **Impacted Entity**, coders selected some stakeholder mentions that lack clear relevance because they do not single out a specific individual or group. For example, coders selected entities such as "our families", "life in our planet", "a person", and "a voter". While these mentions may help the user-community understand how authors express themselves or how authors appeal to the regulatory agencies, we omit such entities from our final definition of a stakeholder.

[3] http://erulemaking.cs.cmu.edu/Data/USEPA-OAR-2002-0056/

Under these more detailed specifications, the second code-book test resulted in the overwhelming majority of **Impacted Entity** mentions in the Mercury Corpus being "women" and "children". It is unclear at this point if such stakeholder mentions are of value to regulatory agencies, as they may already know who the major impacted entities are. Future work will focus on learning from the user community if **Impacted Entity** is a valuable category of stakeholder and, if so, what types of entities should be included. The diversity and value of **Impacted Entity** types might be corpus-specific. Other corpora must be examined in more detail to reach a general conclusion.

This work focuses on learning to automatically extract **Author Self ID** stakeholder mentions, which has, at present, obvious importance and a clearer definition.

3. RELATED WORK

The stakeholder identification approach described in this paper combines two techniques well studied within the language technologies community: template/pattern-based information extraction (IE) and bootstrapping. The goal of pattern-based IE is to learn linguistic expressions that faithfully signal an occurrence of the target class in natural language text. The supervised approach to this problem is to learn these patterns from annotated training data. An alternative that arguably minimizes human effort is bootstrapping. These two techniques have been combined to different extents in previous work.

Hearst [4] describes an algorithm for learning lexico-syntactic patterns that mark relations such as hyponymy. A hyponym relation is an "is-a" relation (e.g., "England" is a hyponym of "country"). The algorithm essentially runs a single bootstrapping iteration. Pairs of entities known beforehand to share some relation (e.g., hyponymy) are collected from a lexical resource, sentences containing both entities are extracted from an unannotated corpus, and lexico-syntactic patterns are learned and used to find new entity pairs. Examples of patterns that mark hyponymy are "countries *such as* England…" or "England *and other* countries…" Nobata & Sekine [6] learns patterns that mark the event of a corporate executive leaving one company for a different one. Like Hearst, Nobata and Sekine starts off with a small set of known true instances of this event. In contrast, they go one step further by clustering extraction patterns and merging patterns belonging to the same cluster. Merging aims to improve recall without sacrificing precision by retaining only the essential constraints implicit in related extraction patterns. In [2], bootstrapping is used to learn extraction patterns for person names, organizations, and locations. However, in their work, the context of an extraction is represented by set of independent features rather than an extraction pattern. Thus, predicting the presence/absence of the target class in a candidate noun-phrase (NP) becomes a classification problem.

Etzioni's KNOWITALL system [3] is a large-scale pattern-based bootstrapping information extraction system intended to populate and extend an ontology or network of relations. A number of pre-set extraction pattern templates (e.g., "NP1 such as NP2", "NP1 including NPList2", and "the NP1 of NP2 is NP3") are used to learn relations, such as `playsFor(Athlete, SportsTeam)`.

The work in this paper is most closely related to that of Riloff et al. [7][8][9][10][11], as will be described in Section 4.

4. ALGORITHM DESCRIPTION

The general structure of our algorithm follows that of [8] and is shown in Figure 1. The goal of the algorithm is to collect a dictionary of extraction patterns and a lexicon of target class instances simultaneously in a bootstrapping framework. Generally, a bootstrapping approach to IE proceeds as follows. The only inputs are a sample of instances of the target class (i.e., seeds) and a large unannotated text corpus. The seeds are used to discover patterns that extract text segments belonging to the target class. Those extraction patterns are used to discover new entities. Those new entities are used to discover new extraction patterns, and so on, for a fixed number of cycles or until some terminating condition is met. In this particular bootstrapping problem, we seek a lexicon of stakeholder NPs and a dictionary of extraction patterns that reliably mark a stakeholder mention.

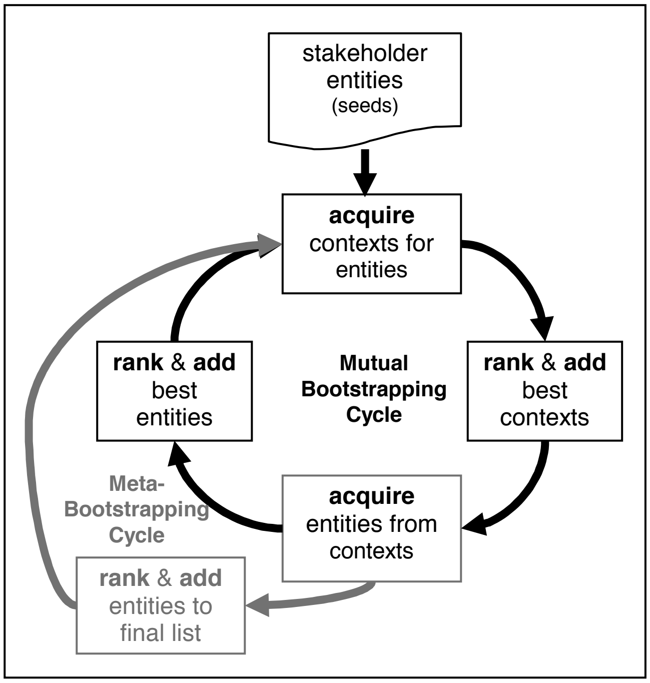
stakeholder entities (seeds)
acquire contexts for entities
rank & add best contexts
acquire entities from contexts
rank & add best entities
Mutual Bootstrapping Cycle
Meta-Bootstrapping Cycle
rank & add entities to final list

Figure 1. Overview of mutual-bootstrapping algorithm

The *inner* cycle is called the *mutual-bootstrapping* cycle. The *outer* cycle is called the *meta-bootstrapping* cycle. The motivation behind a 2-cycle bootstrapping approach, as opposed to a single-cycle approach that monotonically grows the lexicon with more entities is robustness. Any bootstrapping approach is at risk of adding to the lexicon entities that do not belong to the target category. This is particularly true during the first iterations, when the inventory of extraction rules and entities is small. When a bad entity is added to the lexicon, it is used to learn new extraction patterns and to re-weight the extraction patterns already learned. Thus, a critical mass of bad entities in the lexicon can send the bootstrapping algorithm in the wrong direction, meaning that it learns to extract entities that are not of the target class. The (outer) meta-bootstrapping cycle is intended to prevent this from happening by periodically reducing the lexicon to only the best entities and proceeding again with those best entities as the "seed" entities. The mutual-bootstrapping (inner) and meta-bootstrapping (outer) cycles are described in more detail below.

The mutual-bootstrapping (inner) cycle works as follows. The mutual-bootstrapping cycle maintains two lists: a temporary list of stakeholder **e**ntities, `tempEList`, and a temporary list of extraction **p**atterns, `tempPList`. Initially, `tempEList` contains only the stakeholder seeds. During each iteration of the mutual-bootstrapping cycle, new stakeholder entities are added to `tempEList` and new extraction patterns are added to `tempPList` by alternating between two steps. In step one, the union of all extraction patterns co-occurring with entities in `tempEList` is collected. Then, each extraction pattern p_i is scored using the semantic affinity [7][8]

$$score(p_i) = \frac{F_i}{N_i} \cdot \log_2(F_i)$$

where F_i is the number of entities extracted by p_i that are currently in `tempEList` and N_i is the number of all unique entities extracted by pattern p_i. The semantic affinity rates the tendency of pattern p_i to extract entities of the target semantic category. A pattern with a high semantic affinity tends to extract entities that are of the target class (vs. another class) given that a large proportion of the entities it extracts are in `tempEList.`

In step two, the best P extraction patterns are added to `tempPList` and all the entities that were extracted by those new extraction patterns are added to `tempEList`. The mutual-bootstrapping cycle continues alternating between steps one and step two until `tempPList` has *MAX_P* extraction patterns. Then, the algorithm runs one iteration of the meta-bootstrapping (outer) cycle.

In the meta-bootstrapping cycle, all entities are scored and the best E entities not already in the permanent entity list `permEList` are added to `permEList`. The score of entity e_i, shown below, is a function of the number of patterns N_i in `tempPList` that extract e_i. The argument is that an entity that is extracted by more patterns in `tempPList` is more reliable than one that is extracted by only a few. One percent of the score of each pattern p_k is factored into the equation for tie-breaking purposes.

$$score(e_i) = \sum_{k=1}^{N_i} 1 + 0.01 \cdot score(p_k)$$

Upon completing an iteration of the mutual-bootstrapping, the algorithm proceeds with a new round of the mutual-bootstrapping cycles. However, instead of starting with the entities in the seed list, `tempEList` is cleared, `tempEList` is set equal to `permEList`, and the first mutual bootstrapping cycle starts with `tempEList`. The algorithm continues for a fixed number (META) of meta-bootstrapping (outer) iterations.

To complete the description of the algorithm, it is necessary to describe the extraction patterns used. An extraction pattern is an instantiation of an *extraction pattern template*. An extraction pattern template defines the features from the context of the extraction that are used by its extraction patterns. In other words, the extraction pattern template succinctly defines the space of all possible extraction patterns. The goal of a bootstrapping algorithm is to search the space defined by the extraction pattern template for the best extraction patterns.

During evaluation, the meta-bootstrapping algorithm described above was individually combined with 3 different sets of extraction pattern templates. The first two are custom-built templates. Surface-based templates use word surface-form and part-of-speech information. WN-based templates are a more general extension of Surface-based templates and impose a semantic constraint on the extracted stakeholder NP. The third set of extraction templates is the default set of templates available with the Sundance information extraction (IE) engine [9]. These 3 templates are described below.

4.1 Surface-Based Patterns

Surface-based patterns match two words preceding and two words following the stakeholder mention, (i.e., W_{L1} W_{L2} ____ W_{R1} W_{R2}). Each word, W_{XY}, is represented by two features: its surface form and its part-of-speech. In order for a context in the data to match a Surface-based extraction pattern, the word in the data must match the surface-form and part-of-speech of the corresponding W_{XY} in the pattern specification. Two more constraints are imposed. First, the entity encapsulated by words W_{L2} and W_{R1} (the prospective stakeholder) must be no greater than 10 words. Second, the words between W_{L2} and W_{R1} must constitute a noun-phrase. Our implementation uses a greedy NP-chunker that favors short NPs over long NPs. For example, the noun phrase "an avid fisherman and consumer of fish and fish products", is tagged "**an avid fisherman and consumer]$_{\mathbf{NP}}$** of **[fish and fish products]$_{\mathbf{NP}}$**, where **[.]$_{\mathbf{NP}}$** denotes a tagged NP. Rather than requiring the whole stakeholder NP to be tagged as a single NP, we only require the word following W_{L2} to be the first word of an NP and the word preceding W_{R1} to be the last word of an NP. This heuristic is based on the assumption that a disjoint sequence of NPs extracted by a presumably good extraction pattern is in fact a larger NP.

As a pre-processing step, each public comment was sentence segmented, POS-tagged, and NP-chunked. BOS (beginning of sentence) and EOS (end of sentence) tokens were added to the beginning/end of each sentence to allow extraction patterns to encode whether they neighbor the beginning/end of sentence boundary. Punctuation marks were also annotated so they can be treated no different than word tokens. Punctuation marks were not used in Surface-based patterns, but were used in WN-based patterns for reasons stated later. Sentence segmentation was done using LingPipe[4]. POS-tagging and NP-chunking was done using the OpenNLP toolkit[5].

4.1.1 NP-Expansion

An important step in the bootstrapping process is learning new extraction patterns from the current set of stakeholder entities. It is important that the extraction rules learned have high precision so that the process remains focused on the target class, stakeholders. However, recall is also important, otherwise new stakeholder entities are not learned and the bootstrapping process stagnates. Suppose a stakeholder entity such as "a mother" is learned. If this NP frequently occurs as a head noun with modifiers to its right (e.g., a mother **of two** boys, a mother **of three**, a mother **of twins)**, then the learned extraction patterns will suffer from low recall because W_{R1} and W_{R2} will be used to represent those modifiers (shown in bold). Such extraction patterns are not likely to

[4] http://www.alias-i.com/lingpipe/

[5] http://opennlp.sourceforge.net/

generalize well to other stakeholder types. Our solution is NP-expansion by heuristically expanding the NP as much as possible before learning the extraction pattern present in the given context. This does not mean that the stakeholder entity is modified after being added to the list of entities. It means that when a stakeholder entity is used to learn new extraction patterns, if the entity occurs in a context where it can be expanded, it is expanded and the extraction pattern learned is that which surrounds the expanded NP. NP-expansion is achieved by iteratively joining NPs separated by any one of a fixed set of prepositions.[6] For example, if "a mother" is found within the context "As [**a mother**]$_{\textbf{NP}}$ of [**two children**]$_{\textbf{NP}}$ with [**non-verbal disabilities**]$_{\textbf{NP}}$, I think…". Then, the pattern learned is that which surrounds "a mother of two children with non-verbal disabilities". Note that the prepositions "of" and "with" are within the predefined set of prepositions. NP-expansion is done only while collecting extraction patterns. NP-expansion is not done while collecting the entities extractable via the current set of extraction patterns.

4.1.2 Head-NP Querying

As noted above, NPs can be as long as 10 words. An inherent limitation in using longer stakeholder entities to discover new extraction patterns is that longer NPs occur fewer times than shorter ones. To mitigate this sparsity problem of longer stakeholder mentions, only the stakeholder's head NP is used when searching for new extraction patterns. The head NP is heuristically chosen to be the left-most NP. When learning new patterns from a stakeholder, NP-expansion is done on the head NP, just as it would be done on the full stakeholder NP.

4.1.3 List-Handling

A stakeholder mention can occur within a list of stakeholder mentions (e.g., "I am a husband, a father, a teacher, and a concerned North American."). During bootstrapping, one option would be to treat such a list as a single stakeholder mention. A different option would be to split the list into its 4 stakeholder constituents. The heuristic used lies somewhere in the middle. The list is split by collecting all NPs that are separated by a comma and by the conjunction "and". However, splitting on "and" poses a risk. Consider the following statement: "I am a teacher, a civil servant, and I know that it is not easy…". "I", in this case, is an NP, but it is not part of the list of stakeholders, as in the first example. It is difficult to determine whether the NP following the conjunction "and" is a continuation of the list of stakeholders, or if "and" is used to mark the end of the list of stakeholders. The NP following an "and" is heuristically ignored if it is a preposition, such as "I".

4.1.4 Adjective and Adverb Padding

Extraction patterns learned may include nouns and/or verbs (e.g., Surface-based pattern "I/**prp** am/**vbp** _____ ./**period eos**"). To increase the coverage of such a pattern, the pattern is modified to allow an optional adverb wildcard that matches on any token tagged as an adverb before and after the verb. Note that adverbs can occur before and after the verb (e.g., "I am practically a …" and "I practically am a…"). Extraction patterns with a noun are padded with an optional adjective wildcard before the noun. The assumption is that adding these optional adjectives and adverbs-specific wildcards will increase a pattern's coverage without hurting its precision. In other words, adjectives and adverbs are not the key components of the context of a stakeholder mention.

4.2 WN-Based Patterns

The WordNet-based (WN-based) pattern template also departs from observing W_{L1} W_{L2} ____ W_{R1} W_{R2}, two words preceding and two words following the stakeholder mention. For a context in the data to match a WN-based extraction pattern, the word in the data must only match the part-of-speech of the corresponding W_{XY} in the pattern specification, with two exceptions. First, if W_{XY} is a pronoun, then W_{XY} in the text must match both the surface-form and part-of-speech of W_{XY} in the pattern specification. This restriction is imposed to avoid conflating pronouns that refer to the comment's author (e.g., "As a mother of two, *I* think that…") with pronouns that refer to the recipient (e.g., "As someone responsible for the environment, *you* should…") or a third person. Second, if either W_{L1} or W_{L2} (the two words preceding the stakeholder mention) is tagged as a verb, then the word in the text must match both the surface-form and part-of-speech of W_{LY}.

This decision was informed by observing the patterns learned during bootstrapping when this restriction was not made. Without imposing this restriction, a pattern learned is

I/**prp** */**vbp** _____ */**cc** I/**prp**,

where the '*' denotes a POS-specific wildcard operation. This, general pattern subsumes multiple Surface-based patterns. However, it subsumes a combination of good contexts, such as

I/**prp** am/**vbp** _____ and/**cc** I/**prp**

and bad contexts, such as

I/**prp** have/**vbp** _____ and/**cc** I/**prp,**

The second context is bad because the extracted NP does not refer to the author of the public comment (e.g., "I have a year-old son and I want him to…"), so it is not a stakeholder. By imposing the second restriction, WN-based patterns must represent the two contexts above differently, allowing the bootstrapping algorithm the possibility of learning the good one and not the other. More generally, the intuition is the following. If the stakeholder mention immediately follows a verb phrase, it is likely that the stakeholder is the direct object of the verb phrase. The object of some verb phrases (e.g., am, being, having been) will refer to the author of the comment, whereas the object of other verb phrases (e.g., have, having had) will not.

4.2.1 A stakeholder is a hyponym of person

WN-based patterns are more general than Surface-based patterns. The expected recall is higher, possibly at the expense of precision. To avoid hurting precision another constraint is imposed, this time on the stakeholder NP to be extracted by a WN-based pattern. The constraint is that the NP must refer to a person. This constraint was implemented using WordNet. WordNet is a publicly-available lexical database[7]. It is a large network of words (or rather sets of synonyms that refer to the same concept) and relations between those synonym sets. One such relation is that of

[6] of, in, to, for, on, with, at, by, from, as, into, about, like, between, after, through, over, under, against, before, without, within, during, towards, off, upon, including, among, around, across, behind, who

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

hyponymy (the "is a" relation). A doctor is a hyponym of medical practitioner because it is a type of medical practitioner. A doctor is also a hyponym of person because a medical practitioner is a professional and a professional is a person. Thus, the hyponymy relation is not restricted to only a concept's immediate children, but extended to all its descendents. An extracted stakeholder NP must be a hyponym of person in the WordNet hierarchy.

Implementing this constraint poses two challenges, both overcome heuristically. The first challenge is selecting the sense of the prospective stakeholder NP when querying WordNet. A word has multiple senses in WordNet. For example, "MD" is a physician, the radioactive element Mendelevium, the state of Maryland, and a degree in medicine. We heuristically pick the most common sense in WordNet. This is a conservative heuristic. An alternative would be to choose all possible senses. However, that heuristic often fails (e.g., the second most common sense of the noun "bear" is "an investor with a pessimistic market outlook" and is a hyponym of person). Another alternative is to adopt a conservative heuristic during training (bootstrapping) and a more relaxed heuristic during testing. The second challenge is that WordNet cannot be queried with an NP such as "a concerned citizen with a degree in environmental science" as it will not be found in the database. Thus, the head noun, in this case "citizen", must be used to query WordNet. As explained in Section 4.1, the extracted NP is possibly a disjoint sequence of tagged NPs, in this case "**[a concerned citizen]**$_{\mathbf{NP}}$ with **[a degree in environmental science]**$_{\mathbf{NP}}$". The head noun is assumed heuristically to be within the left-most NP, "**[a concerned citizen]**$_{\mathbf{NP}}$". If this left-most NP is composed of words $w_{0,}w_1,...,w_n$, WordNet is first queried using all words. If the NP is not found, then WordNet is queried using words $w_1,...,w_n$, iteratively omitting the left-most word until the NP is found in WordNet. If, at worst, w_n is not found in WordNet, then the NP is not subsumed by person and the NP is not extracted as a stakeholder. The major assumption of this heuristic is that the left-most NP will contain the head noun and that any NPs to the right of this left-most NP will simply modify it, as in the case of "**[a scientist and engineer]**$_{\mathbf{NP}}$ in **[the energy field]**$_{\mathbf{NP}}$". This heuristic may fail, as in the case of "**[a child development]**$_{\mathbf{NP}}$ and **[health care specialist]**$_{\mathbf{NP}}$". However, as will be shown, such cases are the exception rather than the rule.

Table 1 summarizes the different features and heuristics used with Surface- and WN-based patterns.

Table 1. Summary of Surface-based & WN-based features.

	Surface-based Patterns	**WN-based Patterns**
NP-Expansion	Yes	Yes
$\boldsymbol{W_{XY}}$ **must match surface-form & POS**	Yes	No
Stakeholder is hyponym of person	No	Yes
Head NP Querying	Yes	Yes
List-Handling	Yes	Yes
Adj./Adv. Padding	Yes	Yes
Punctuation	No	Yes

4.3 Sundance-Based Patterns

Sundance is a shallow parser built at the University of Utah [9]. Built on top of Sundance is an information extraction engine that can be used to facilitate information extraction on a new text corpus. The IE engine automatically generates all extraction patterns for the new corpus based on a finite set of pre-defined extraction pattern templates. The Sundance IE engine exhaustively applies all these extraction patterns to the corpus. The output is all NPs extractable via any Sundance-based pattern. This set of NPs constitutes most NPs in the corpus. Some NPs are not extractable, for reasons noted later.

The extraction patterns produced for the new corpus are all instantiations of the 17 extraction pattern templates made available with the Sundance suite of tools. We chose to use these 17 extraction pattern templates because they have been applied successfully in various information extraction tasks, such as creating objective/subjective sentence classifiers [10], extracting opinion-holders from opinion-sentences [1], and learning semantic lexicons of entity types such as *target* and *victim* [5] *building name*, *event*, *human*, *location*, *time*, and *weapon* [11], and *company name*, *location*, *professional title*, and *weapon* [8]. Table 2 shows the 17 extraction pattern templates and, for each template, an actual sample instantiation from applying the Sundance IE engine to the EPA's Mercury corpus. In both pattern templates and pattern instantiations, the item enclosed in '<' and '>' is the slot filled by the extraction, the predicted stakeholder.

Table 2. Sundance extraction pattern templates and sample instantiations.

#	**Extraction Pattern Template**	**Instantiation**
1.	<subj> AuxVp Dobj	<subj> makes decisions
2.	<subj> AuxVp AdjP	<subj> is concerned
3.	<subj> PassVp	<subj> was distributed
4.	<subj> ActVp	<subj> requires
5.	<subj> ActVp Dobj	<subj> takes care
6.	<subj> ActInfVp	<subj> continues to ignore
7.	<subj> PassInfVp	<subj> was formed to protect
8.	ActVp <dobj>	forget <dobj>
9.	Subj AuxVp <dobj>	technology is <dobj>
10.	ActInfVp <dobj>	likes to add <dobj>
11.	PassInfVp <dobj>	is required to furnish <dobj>
12.	InfVp <dobj>	to show <dobj>
13.	ActVp Prep <np>	demand as <np>
14.	InfVp Prep <np>	benefit from <np>
15.	PassVp Prep <np>	is regulated through <np>
16.	Np Prep <np>	representative of <np>
17.	Np has <possessive>	<np>'s effects

The following observation on these extraction pattern templates motivates our choice to experiment with Sundance-based extraction patterns for this particular task. First, 15 of the 17 patterns are verb-centric. Pattern templates 1-15 will produce extraction patterns that, if effective in extracting stakeholders, will be because stakeholder mentions tend to occur in the discourse as arguments of certain verbs. Specifically, templates 1-7 will produce effective extraction patterns if it is true that stakeholders tend to occur as subjects of certain verb-types. Templates 8-12 will be effective if stakeholders tend to be in the direct object position of certain verb-types. Likewise, 13-15 will be effective if stakeholders tend to occur in prepositions that attach to certain verb-types. These verb-centric pattern templates have been successful in previous information extraction tasks. In [8], among the top 20 most successful patterns for extracting *company names* were "<subj> employed" and "owned by <dobj>", and effective patterns for *location names* were "living in <dobj>" and "traveled to <dobj>". Our research investigates whether it is possible to distinguish between **Author Self ID** stakeholder mentions and non-target-class noun phrases based only on the verbs that such NPs occur as arguments of.

5. EVALUATION

5.1 Evaluation Methodology

In all experiments, the EPA's Mercury Corpus was used for (unsupervised) training and testing. As mentioned previously, the approximate 500,000 documents in the Mercury Corpus are a combination of completely original public comments, exact duplicates, and modified form letters. Duplicate text, either in a comment that is an exact duplicate of another or in the duplicate portion of a modified form letter, is undesirable during both training and evaluation because it artificially inflates NP frequency counts without providing new information. Thus, all duplicate text was removed using the Durian duplicate detection tool [14]. The result is a set of about 120,000 completely original or modified form-letter comments. We train the meta-bootstrapping algorithm only on non-duplicate text with no stakeholder annotations. The stakeholder-annotated test set is also comprised of only non-duplicate text.

Evaluation was conducted on a set of 1,020 public comments from the Mercury Corpus. The full 1,020 document set was annotated by a single coder following the coding scheme described in Section 2. The reliability of this single coder's annotations was evaluated using inter-coder agreement with respect to a second coder's annotation of 510 of the 1,020 documents, 50% of test collection. Agreement between the two coders was measured using f-measure, considering one coder's annotations as the "gold-standard" and the second coder's annotations as the "predictions". The f-measure, considering only agreements between coders that are an exact match, was 0.53. The f-measure (overlap), which considers correct all agreements between coders with word overlap, was 0.70. The number of **Author Self ID** stakeholder mentions found in the 1,020 document test set was 60. In the 510 document portion used to test inter-coder agreement (coded by both coders), the coder who coded the full set found 43 **Author Self ID** mentions. The second coder found 37. Thus, using the coding scheme described in Section 2, a human is expected to find 1 **Author Self ID** in about every 20 documents.

Given the relatively small amount of human-annotated data (1,020 documents), all parameters in the meta-bootstrapping algorithm were set using only the unannotated training set, by examining the quality of extractions made by the bootstrapping algorithm on the training set. The annotated test set was not used in parameter tuning. The parameter values were held constant across all experiments. The parameters were set as follows. After each iteration of the mutual-bootstrapping (inner) cycle, the best 5 extraction patterns not already in `tempPList` were added to `tempPList` (*P*=5). A meta-bootstrapping (outer) cycle was run after adding a total of 80 extraction patterns to `tempPList` (*MAX_P* = 80). After each mutual-bootstrapping (outer) cycle, the best 40 stakeholder entities not already in `permEList` were added to `permEList` (*E*=40). The algorithm was run for 4 mutual-bootstrapping (outer) cycles (*META*=4).

During each evaluation, the 80 extraction patterns in `tempPList` after the last iteration of the mutual-bootstrapping (inner) cycle were applied to the test set and those extracted noun phrases were compared to the gold-standard stakeholders in terms of precision, recall, and f-measure. Precision, recall, and f-measure were calculated under two criteria. Under the first criteria, called "inclusive", a predicted stakeholder noun phrase is correct only if it is fully contained in the reference stakeholder noun phrase, or vice versa. Under the second criteria, called "exact", a predicted stakeholder noun phrase is correct only if it exactly matches the reference stakeholder noun phrase. Results are presented in terms of the six metrics: P_{inc}, R_{inc}, $F_{inc.}$, P_{exact}, R_{exact}, and F_{exact}, where "inc" stands for "inclusive". When applying the learned extraction patterns to the test set, it is possible that several extraction patterns will extract different extents of the same stakeholder NP (e.g., "a citizen" and "a citizen of this country"). In such cases, we discard all but the longest stakeholder NP, irrespective of whether it is a true positive or a false positive. On average, this does not bias any of the metrics above because the coder did not always choose the longest NP as the stakeholder. It is a heuristic applied consistently on all extractions on all experiments.

5.2 Evaluation of Surface- and WN-based patterns

The following experiment was run by seeding the system with 5 seeds: **a biologist**, **an environmentalist**, **a resident**, **a citizen**, and **an American**.

The two-cycle bootstrapping algorithm was run using Surface- and WN-based pattern templates individually. Evaluation results are shown in Table 3. .

Table 3. Results for Surface-Based (SURF) Patterns and WN-based (WN) Patterns.

	P_{inc}	R_{inc}	F_{inc}	P_{exact}	R_{exact}	F_{exact}
SURF	.590	.383	.474	.487	.317	.384
WN	.549	.650	.595	.394	.467	.427

WN-based patterns achieved a higher f-measure (inclusive and exact) than Surface-based patterns. R_{inc} is about 70% higher and only about 7% lower for WN- vs. Surface-based patterns. In terms of R_{exact} and P_{exact} the improvement in recall is about 45% and the loss in precision is about 20% for WN- vs. Surface-based pattern.

Examining the extraction patterns learned via Surface-based patterns reveals one reason why it obtains lower recall. Table 4 shows the 10 most highly weighted extraction patterns learned using Surface-based patterns. "**bos**" denotes the beginning of sentence marker.

Table 4. The 10 most highly weighted Surface-based patterns[8].

Surface-based Patterns
bos for/**in** _____ that/**wdt** is/**vbz**
bos as/**in** _____ who/**wp** work/**vbz**
bos as/**in** _____ who/**wp** sees/ **vbz**
bos as/**in** _____ who/**wp** s/**vbz**
bos as/**in** _____ who/**wp** plans/**vbz**
bos as/**in** _____ who/**wp** lives/**vbz**
bos as/**in** _____ who/**wp** is/**vbz**
bos as/**in** _____ who/**wp** has/**vbz**
bos as/**in** _____ who/**wp** believes/**vbz**
bos as/**in** _____ who/**wp** worked/**vbd**

These extraction patterns indicate that the algorithm became stuck in a potentially sub-optimal state. In other words, the algorithm found a very good pattern "**bos** As ____ who */**verb**", and focused its efforts in finding instantiations of this pattern with different verb-forms (e.g., "work", "sees", "plans", "lives", "is", "has", "believes") when, in fact, perhaps any verb suffices. Using WN-based patterns, the algorithm learns

bos */**in** _____ */**wp** */**vbz**.

This pattern subsumes those patterns described above, extracting all stakeholders predicted by those patterns and a few more.

A natural question is whether the "is a person" constraint imposed by WN-based patterns on extracted stakeholder NPs is effective. We investigated how many true, reference, stakeholders would be missed because they do not occur in WordNet or because the heuristic for finding the stakeholder's head noun (See 4.2.1) is too naïve. The fraction of reference stakeholders that do pass the "is a person" constraint imposed by WN-based patterns is an upper bound on recall. Of the 60 reference stakeholders, 9 do not pass the "is a person" test. One mistake is made because the most common sense of the stakeholder's head noun ("holder", as in "the holders of an annual national park pass"), is not the sense of holder that is subsumed by "person". One mistake is made because the head-noun ("homeschooler") is not in WordNet. The rest of the mistakes are due to NP-chunking mistakes. For example, "a life-long card carrying check writing Republican" is tagged **"[a life-long card]$_{NP}$** carrying **[check writing Republican]$_{NP}$**" and "card" is not subsumed by person. The best attainable recall using WN-based patterns is 85%.

The next question to explore is whether the "is a person" constraint allows the learning of valuable patterns that would not have been learned otherwise or may not be effective without this constraint. As mentioned previously, an important pattern learned using WN-based extraction patterns is **bos** */**in** _____ I/**prp** */**vbz**

This pattern extracts 7 of the 60 reference stakeholders. Results show that the "is a person" constraint prevents 4 false hits that occur in contexts that would match this pattern. One such case is in "As a result I have Fibromyaglia…". Another case is "As the federal agency responsible for regulating environmental impacts from industry I am severely disappointed that …". Ironically, in the second case the NP refers to the recipient of the comment instead of the author, in spite of the NP being immediately followed by "I".

As mentioned in Section 4.1, punctuation was ignored while running the bootstrapping algorithm with surface-based patterns. This decision was made because ignoring punctuation improves f-measure (inclusive & exact) for surface-based patterns, increasing the baseline for comparison with WN-based patterns. Results for surface-based patterns when punctuation is not ignored are shown in Table 5.

Table 5. Experimental results for Surface-Based Patterns without ignoring punctuation.

P_{inc}	R_{inc}	F_{inc}	P_{exact}	R_{exact}	F_{exact}
.260	.733	.384	.178	.500	.262

Incorporating punctuation lowers the precision of surface-based patterns because the patterns learned are less constrained in the sense that they convey less meaningful information about the context of the extraction. For example, consider the context "**bos** as/**in** ____ ,/**comma** I/**prp** */**vbp**". When punctuation is ignored, the surface-based pattern learned is "**bos** as/**in** ____ I/**prp** */**vbp**". When punctuation is not ignored, the pattern learned is "**bos** as/**in** ____ ,/**comma** I/**prp**", which is too unconstrained. Given less constrained patterns, the algorithm drifts away from the target class and essentially learns to extract noun-phrases. As evidence, the most highly weighted surface-based pattern learned when punctuation is not ignored is ",/**comma** and/**cc** ____ ./**period eos**", where "**eos**" is the "end of sentence" marker. This pattern extracts 135 incorrect stakeholder entities in the test set.

5.3 Evaluation of Sundance-Based Extraction Patterns

To evaluate Sundance-based patterns, the training and test collections were first run through the Sundance information extraction (IE) engine. The Sundance IE engine takes as input the set of 17 default extraction templates (see Table 2, column 1) and finds all instantiations (called "patterns" from hereon) of these pattern templates (see Table 2, column 2) in the text. The Sundance IE engine annotates each NP in the corpus with the pattern instantiation that extracts it (if one exists). This output was then applied to the bootstrapping algorithm.

A fair evaluation of Sundance-based patterns requires carefully selecting the stakeholder seeds because an occurrence of a seed in the data requires an exact match between the seed NP and the NP annotated by the Sundance IE engine. Thus, the system was seeded with all NPs annotated by the Sundance IE engine that fully contain any of the seeds listed in Section 5.2. Other than using a larger set of seeds, no other bootstrapping parameter was changed. Although some of the extraction patterns learned by the algorithm seem reasonable (e.g., worry as <np>, <subj> is concerned, <subj> is interested), none of the learned patterns occur in the test set. Thus, an alternative evaluation approach was adopted for Sundance-based patterns. The test set was inspected and the bootstrapping algorithm was seeded with entities that should bias it to learn patterns that do occur in the test set. A description of the steps taken follows.

[8] **bos** = beginning of sentence, **in** = preposition, **wdt** = wh-determiner, **vbz** = verb, 3rd person singular, present, **vbd** = verb, past.

First, the output of the Sundance IE engine on the test set was examined to collect the full set of extraction patterns that extract at least one true stakeholder on the test set. Each of these patterns was evaluated in isolation in terms of P_{inc}, R_{inc}, and F_{inc}. Results for the 5 extraction patterns with the highest recall are shown in Table 6. Of the patterns in Table 6, the best pattern, with significantly higher F_{inc}, is "I am <dobj>" (F_{inc} = .316).

Table 6. Sundance-based patterns with highest recall

Pattern	P_{inc}	R_{inc}	F_{inc}
I am <dobj>	.316	.316	.316
serving as <dobj>	.500	.017	.032
<subj> known	.059	.017	.026
worked as <np>	.500	.017	.032
<subj> pass	.059	.017	.026

Next, the bootstrapping algorithm was run by seeding the system with the 1583 noun-phrases extracted from the training set by this pattern. By doing so, the bootstrapping algorithm is biased to learn this pattern, which would help it perform well on the test set. The accumulated set of extraction patterns learned after each mutual-bootstrapping iteration were evaluated on the test set in terms of P_{inc}, R_{inc}, and F_{inc}. Figure 2 shows these results. Iterations 17 and 22, marked with a dotted line, correspond to the first mutual-bootstrapping (inner) cycle after a meta-bootstrapping cycle. The increase in precision is due to the system restarting with only the best entities as seeds.

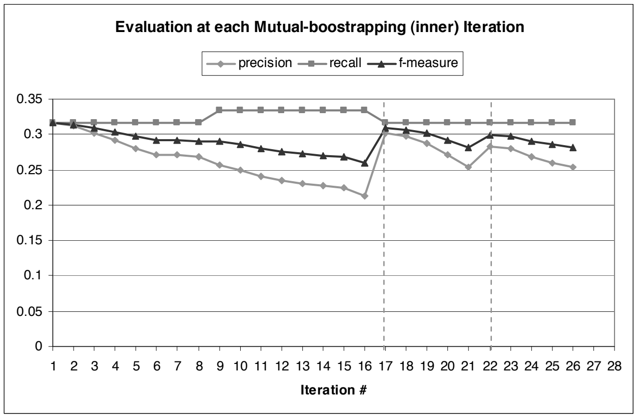

Figure 2. P_{inc}, R_{inc}, F_{inc} on test set after each iteration of mutual-bootstrapping (inner) iteration.

By seeding the system with entities extracted by the pattern "I am <dobj>", the algorithm learns this pattern in the first mutual-bootstrapping cycle and obtains a recall of .316 on the test set. The pattern remains highly weighted throughout the whole process, so the recall does not drop below .316. Although, the bootstrapping algorithm does not drift away from the target class enough to remove this good pattern from its list, the algorithm fails to learn more extraction patterns that are effective on the test set. Precision consistently degrades between subsequent mutual-bootstrapping (inner) iterations and recall remains mostly constant. Interestingly, the bootstrapping algorithm does not complete all mutual-bootstrapping and meta-bootstrapping iterations, meaning that it reaches the point where no new patterns/stakeholders are learned given the current inventory of stakeholders/patterns. One possible reason is that the NPs extracted by the Sundance IE engine are too long and therefore infrequent in the data, in which case heuristics like np-expansion, head-np querying, and list-handling (see Section 4.1) might help.

After inspecting the output of the Sundance IE engine on the test set, it was discovered that the highest attainable recall on the test set using Sundance-based patterns is 0.617. That is, approximately 40% of the true stakeholders in the test set were not extracted by any Sundance-based pattern instantiation. A major reason for this is that many stakeholders occur in contexts where pronoun resolution is required in order to extract the entity with a Sundance-based pattern. The most common context surrounding true stakeholders in the test set is "**bos** As ____ I", which alone obtains a P_{inc}, R_{inc}, and F_{inc} of 0.567, 0.350, and 0.433, respectively. Thus, at least 35% of the true stakeholders in the test set appear in a context where pronoun resolution is required in order for the stakeholder to be extracted via a Sundance-based pattern.

The Sundance IE engine handles pronoun resolution via at least two heuristics. In the first scenario, the Sundance parser does not resolve "I" and so "I" becomes the extracted entity, as in

> "As a former employee in the power industry I know there …"

Extraction pattern "<subj> know" extracts "I" instead of "a former employee in the power industry" as the subject of "know". In the second scenario, the Sundance IE engine assumes that the closest noun phrase to the left of the pronoun is its referent. In

> "As a woman of child-bearing age, I now have an additional worry of whether my …"

the Sundance IE engine applies pattern "<subj> have worry" and selects "child-bearing age" as the referent of "I". If pronoun "I" is not resolved, the bootstrapping algorithm misses an opportunity to learn new stakeholders. If "I" is resolved to the wrong entity, the algorithm proceeds with some entities that are not stakeholders and is more prone to drift away from the target class.

6. CONCLUSION

We investigated stakeholder identification in public comment corpora, where a stakeholder is defined as a noun phrase that identifies a group or community of which the comment's author is a member. An existing bootstrapping algorithm is individually combined with 3 different extraction pattern templates. The highest performance in terms of f-measure is obtained using WordNet-based (WN-based) patterns. WN-based patterns achieve higher recall by over-generalizing from the extraction's immediate context and avoid a loss in precision by imposing a semantic constraint on the extracted noun phrase. Surface-based and WN-based pattern outperform Sundance-based patterns. Sundance-based patterns capitalize on the fact that noun phrases belonging to the target class (e.g., stakeholders) appear as an argument (e.g., subject, direct object) of some verbs more often than others. Results suggest that this constraint is not enough to separate stakeholders from non-stakeholders. Also, a challenge in applying Sundance-based patterns to this domain is that pronoun resolution, which is prone to error, is often required.

The outcome of this investigation is a system that automatically identifies the groups and communities represented in a large corpus of public comments. The goal of this novel technology within digital governance is to assist regulatory agencies that need to quickly and thoroughly understand the major concerns of those affected by new regulation. Towards that end, automatically extracted stakeholders provide a quick overview of the communities that authors express membership in. Stakeholders from the mercury corpus include 899 mothers, 64 chemists, 14 neurobiologists, 8 toxicologists, 10 dentists, 3 Minnesotans, and 1 retired senior vice president of a Fortune 100 company. Extracted stakeholders can provide a browsable index for someone to navigate a corpus more productively, for example by locating comments from underrepresented stakeholder groups or authors with technical expertise.

Although our results are promising, about 40% of the stakeholders identified manually are being missed by the algorithm, so there is room for improvement. There are at least two reasons why an effective extraction pattern may not be learned. First, rare patterns pose a challenge. Even if the pattern's precision is high, its low recall will make the bootstrapping algorithm ignore it. Effective extraction patterns follow a heavy-tailed distribution. A few patterns extract many stakeholders and many patterns extract only a few stakeholders. The challenge of extracting high-precision rare patterns is one of representation. Irrelevant features within the extraction's immediate context (e.g., two words before/after it) should be ignored in the presence of more meaningful long-range evidence. A pattern with more meaningful features may be more frequent and is more likely to be learned. Second, some contexts are bound to be ambiguous (i.e., surround stakeholders and non-stakeholders alike) regardless of representation. This problem may be mitigated by applying constraints on the extraction itself, rather than the context, as it is done with WN-based patterns.

In addition to improving the quality and quantity of extractions, future work in stakeholder identification might consider the problem of how to arrange extracted stakeholders so that a person can use them to navigate a corpus. Using WN-based patterns, the bootstrapping algorithm finds 12,249 stakeholders in the Mercury corpus. Given this many stakeholders, it is necessary to organize them. Furthermore, stakeholder identification may be a useful component for other text mining applications, such as relation mining, identifying constructive, well-informed opinions, sentiment analysis, and for summarizing the opinions of broad stakeholder communities.

7. ACKNOWLEDGEMENTS

This research was supported by NSF grant IIS-0429102. We are grateful to the US EPA for providing the public comment data that made this research possible. We thank Stuart Shulman and the Qualitative Data Analysis Program (QDAP) for coding and assisting in preparing the data used in this evaluation. The coding effort was funded by NSF grant IIS-0429293, awarded to the University of Pittsburgh. We are also grateful for the comments from the anonymous reviewers. Any opinions, findings, conclusions, or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In *Proceedings of the Human Language Technology Conference / Conference on Empirical Methods in Natural Language Processing (HLT\EMNLP-05)*. 2005.

[2] Collins, M. and Singer, Y. Unsupervised models for named entity Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-99)*. 1999.

[3] Etzioni, O., Kok, S., Soderland, S., Caferella, M., Popescu, A.M., Weld, D., Downey D., Shaked, T., Yates, A. Web-Scale Information Extraction in KnowItAll (Preliminary Results). In *Proceedings of the World Wide Web Conference (WWW-04)*. 2004.

[4] Hearst, M. Automatic Discovery of WordNet Relations. In *WordNet: An Electronic Lexical Database and Some of its Applications*. Christiane Fellbaum, MIT Press, 1998.

[5] Kwon, N., Shulman, S., and Hovy, E. Multidimensional Text Analysis for eRulemaking. In *Proceedings of the Sixth National Conference on Digital Government Research (dg.o-06)*. 2006.

[6] Nobata, C. and Sekine S. Towards Automatic Acquisition of Patterns for Information Extraction. In *Proceedings of the International Conference of Computer Processing of Oriental Languages*. 1999.

[7] Patwardhan S. and Riloff, E. Learning Domain Specific Information Extraction Patterns from the Web. In *Proceedings of the Workshop on Information Extraction beyond the Document (ACL-06)*. 2006.

[8] Riloff, E. and Jones, R. Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*. 1999.

[9] Riloff, E. and Phillips, W. *An Introduction to the Sundance and Autoslog Systems*. University of Utah Technical Report #UUCS-04-015. 2004

[10] Riloff, E. and Weibe, J. Learning Extraction Patterns for Subjective Expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP-03)*. 2003.

[11] Thelen, M. and Riloff, E. A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP-02)*. 2002.

[12] Yang, H. and Callan, J. Near-duplicate Detection for eRuleMaking. In *Proceedings of the 5th National Conference on Digital Government Research (dg.o-05)*. 2005.

[13] Yang, H., Callan, J., and Shulman, S. Next Steps in Near-Duplicate Detection for eRulemaking. In *Proceedings of the Sixth National Conference on Digital Government Research (dg.o-06)*. 2006.

[14] Yang, H. and Callan, J. Near-duplicate Detection by Instance-level Constrained Clustering. In *Proceedings of the 29th ACM Conference on Research and Development in Information Retrieval (SIGIR-06)*. 2006